Universidade do Minho
Escola de Engenharia

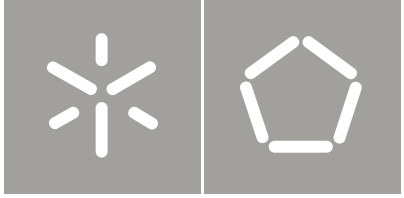
Fábio Barbosa de Lima

Formato de Representação de Eventos de Segurança de Informação

Janeiro de 2014

Fábio Barbosa de Lima Formato de Representação de Eventos de Segurança de Informação

UMinho | 2014



Universidade do Minho
Escola de Engenharia

Fábio Barbosa de Lima

Formato de Representação de Eventos de Segurança de Informação

Dissertação de Mestrado
Engenharia e Gestão de Sistemas de Informação

Trabalho efetuado sob a orientação do
Professor Doutor Henrique Manuel Dinis dos Santos

Janeiro de 2014

DECLARAÇÃO

Nome: Fábio Barbosa de Lima

Endereço eletrónico: pg19680@alunos.uminho.pt Telefone: 912724031

Número do Bilhete de Identidade: 12719431

Título dissertação □/tese □

Formato de Representação de Eventos de Segurança de Informação

Orientador:

Professor Doutor Henrique Manuel Dinis dos Santos

Ano de conclusão: 2014

Designação do Mestrado:

Mestrado em Engenharia e Gestão de Sistemas de Informação

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respectiva, deve constar uma das seguintes declarações:

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, ____/____/_____

Assinatura: _____

Agradecimentos

Esta dissertação não seria possível de realizar sem a compreensão, motivação e colaboração de algumas pessoas. A ordem de agradecimentos é irrelevante, na medida que todos os intervenientes têm um especial lugar de destaque.

Quero prestar os meus agradecimentos ao professor e amigo Henrique Santos, sobretudo pelas orientações ao longo desta dissertação, pela paciência e pelo contributo precioso na ultrapassagem das adversidades que foram surgindo, sendo uma figura sempre presente. A energia e entusiasmo que traspassa são contagiantes, revelando-se uma enorme e verdadeira fonte de inspiração e motivação.

À minha família, em particular aos meus pais e irmã pelos valores que me definem e pelo carinho, coragem e incentivo dado, para que finaliza-se com sucesso mais uma etapa. Sempre presentes são a minha pedra basilar.

Aos meus amigos em geral, pela compreensão nas minhas longas ausências, contudo mantendo-se sempre fiéis e presentes. Em especial à Ângela, à Cláudia e ao Sidónio pelo importante significado que cada um teve à sua especial maneira, no desenrolar desta importante etapa. Somos Bons.

De âmbito institucional, quero deixar uma palavra de apreço à Universidade do Minho, por disponibilizar o espaço e ferramentas necessárias para o desenvolvimento desta dissertação e também à *TRW Automotive* Ponte de Lima, pela flexibilidade e compreensão neste período.

Esta página foi colocada propositadamente em branco.

Resumo

Nos últimos anos, o crescimento da utilização das Tecnologias e Sistemas de Informação nas organizações, aliado ao aumento da dependência da Internet, trouxe consigo um conjunto de ameaças que comprometem os seus Sistemas de Informação (SI). A propagação dessas ameaças tem aumentado, quer em número quer em sofisticação, a um ritmo alarmante, e desperta a atenção de todos aqueles que querem proteger e zelar os seus sistemas, sendo esta uma prioridade das organizações.

De forma a colmatar este problema, existem várias abordagens para prevenir e detetar as ameaças aos SI, contudo as abordagens existentes por si só, não são suficientes para responder às necessidades reais do problema.

Uma crescente solução é a Gestão dos *Logs*, especificamente a análise dos eventos enquanto técnica/ferramenta fundamental na deteção de falhas do sistema e da rede e também na deteção e prevenção de atividades que colocam em causa os SI. O contributo que os eventos podem dar é extremamente importante no auxílio e orientação no combate às ameaças aos seus SI. Contudo, a exploração dos *logs* não é uma tarefa fácil/trivial, devido à heterogeneidade e dispersão dos eventos e pela inconsistência do seu conteúdo e formato.

Os *logs* devem ter um formato padronizado de modo a que se possa tirar partido das suas potencialidades de forma eficiente e inteligente.

Nesta dissertação propõe-se um formato de representação de dados adequado a uma gestão de eventos de segurança de informação integrada e uma *interface* capaz de transformar os dados obtidos em informação útil, a partir de diversos sistemas de registos *logs*.

Palavras-chave: Formatos, Eventos de Segurança de Informação, *Logs*

Esta página foi colocada propositadamente em branco.

Abstract

In the last years, the growth and use of Technology and Information Systems by organizations, ally with the increasing dependence of the internet, provides a group of threats that compromise it Information Systems (IS). The spread of those threats it's concerning and captures the attentions of those who pretend to protect and ensure there systems, becoming a priority to every organization. It's one of the main priorities of every organization.

In order to solve this problem, there are many approaches to prevent and detect the IS's threats. Although there are many solutions they don't are enough to solve the raised problem.

A growing solution its Logs Management, specifically the events analysis as fundamental technique/tool to detect network and systems flaws also detect and prevent activities that compromise IS. The contribution of these events it extremely important to support and guide against the IS threats in an intelligence way. However, the logs exploration it's not an easy task to do, result from the heterogeneity and dispersion of the events, and still the inconsistency of its contents and form.

The logs must have a standard form in order to take advantage of its potentialities, in an efficient and intelligent way.

In these master's thesis its proposed a representation form of data suitable to a management of integrated information security events and an interface capable of transform the obtained data in useful information, through various log record systems.

Keywords: Formats, Security Information Events, *Logs*

Esta página foi colocada propositadamente em branco.

Índice

Acrónimos

Acrónimo	Descrição
APIs	*Application Programming Interfaces*
CBE	*Common Base Event*
CDET	*Common Dictionary Events Taxonomy*
CIA	*Confidentiality, Integrity and Availability*
CIDF	*Common Intrusion Detection Framework*
CEE	*Common Event Express*
CEF	*Common Event Format*
CELR	*Common Events Log Recomendations*
CISL	*Common Intrusion Specification Language*
CLF	*Common Log Format*
CLS	*Common log sintax*
CSV	*Comma Separated Values*
CSIRTs	*Computer Security Incident Response Teams*
CSRG	*Computer Science Research Group*
DARPA	*Defense Advanced Research Projects Agency*
DDoS	*Distributed denial of service*
DoS	*Denial of service*
ECLF	*Extended Common Log Format*
ELF	*Extended Log Format*
HIDS	*Host-based Intrusion Detection System*
HTTP	*Hypertext Transfer Protocol*
IDMEF	*Intrusion Detection Message Exchange Format*
IDS	*Intrusion Detection System*

IDSC	*Intrusion Detection Systems Consortium*
IEEE	*Institute of Electrical and Electronics Engineers*
IODEF	*Incident Object Description Exchange Format*
IP	*Internet Protocol*
IPFIX	*Internet Protocol Flow Information Export*
IPS	*Intrusion Prevention Systems*
JSON	*JavaScript Object Notation*
LEEF	*Log Event Extended Format*
NCSA	*National Center for Supercomputing Applications*
NIDS	*Network-Based Intrusion detection system*
NIST	*National Institute of Standards and Technology*
OSSIM	*Open Source Security Information Management*
SDEE	*Security Device Event Exchange*
SEM	*Security Event Management*
SGML	*Standard Generalized Markup Language*
SI	Sistemas de Informação
SIEM	*Security Information and Event Management*
SIM	*Security Information Management*
SNMP	*Simple Network Management Protocol*
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*
SSL	*Secure Sockets Layer*
W3C	*World Wide Web Consortium*
WELF	*WebTrends Enhanced Log File Format*
XDAS	*Distributed Audit Service*
XML	*Extensible Markup Language*

Índice de Figuras

Índice de Tabelas

1. Introdução

A informação é um dos ativos mais importantes de qualquer organização. Portanto, proteger e garantir a segurança desta força vital é extremamente importante, e deve ser uma das suas principais prioridades.

A segurança da informação não é apenas um simples controlo de utilizadores e palavras-passes. Os profissionais desta área necessitam de informação detalhada sobre a melhor forma de proteger, identificar e prevenir os ataques aos SI. Ameaças contra os ativos de informação, incluem ataques de código malicioso (como a execução de vírus, *worms*, cavalos de tróia e *web scripts*) com o intuito de destruir ou roubar informação, bem como ataques por *denial-of-service* (DoS) e *distributed denial-of-service*, *spoofing*, *man-in-the-middle*, *snnifers*, entre outros. Estas ameaças crescem a um ritmo estonteante e são cada vez mais comuns, complexas e lesivas o que leva às organizações a preocuparem-se com a segurança da informação. As organizações adotam e implementam diversas tecnologias e sistemas de segurança como por exemplo os antivírus, redes privadas virtuais, *firewalls*, entre outros.

Infelizmente não existe uma fórmula única que possa garantir na sua totalidade a segurança da informação. Recentemente tem ocorrido alguns ataques que têm tido um grande impacto nos SI das organizações. Enumeram-se alguns casos:

- **Operação Aurora** – Este ataque designado de "Operação Aurora" foi um ataque coordenado que incluiu um código de programação que explora uma vulnerabilidade do *internet explorer* com o intuito de obter acesso a computadores. Foi projetado para infetar máquinas, ocultar o acesso e roubar ou modificar dados sem deteção. A Google e pelo menos 20 outras empresas foram alvos deste ataque (McAfee, 2010).
- **RSA Security** – A conceituada empresa de segurança americana também foi alvo de ataques. Em carta aberta, o presidente executivo da RSA informou que esses ataques resultaram na extração de algumas informações relacionadas com a RSA *SecurID*, que podem afetar os seus clientes. Esses ataques levaram à substituição de praticamente todos os 40 milhões de *tokens SecurID* (Coviello, 2011).

- **Coca-Cola** – No meio de um acordo de aquisição entre a Coca-Cola e a China Huiyuan Juice Group, que envolve uma tentativa de aquisição na ordem dos 2,4 bilhões dólares, informações confidenciais sobre o negócio são roubadas por *hackers*, e ditam o fracasso deste negócio. Cada vez mais, esses tipos de ataques têm como alvo não apenas informa9ções pessoais, mas também negócios e fusões comerciais (Elgin, B., Lawrence, D., & Riley, 2012).
- **Paypal** – O ataque realizado ao serviço Paypal tem como objetivo passar a segurança dos *sites* contra os ataques pela técnica de *pishing*. Este ataque levou à perda na ordem dos 4.700.000 euros (SOL, 2012).

Deste modo, devido ao sucesso dos ataques enunciados anteriormente, e segundo a literatura atual, é evidente que as tecnologias de segurança utilizadas tradicionalmente nas organizações, não são totalmente capazes de identificar e prevenir da melhor forma os constantes e sofisticados ataques. É necessário explorar métodos mais sofisticados que sejam eficazes na deteção e combate de comportamentos mal-intencionados.

Uma importante fonte de informação no auxílio da segurança de informação são os eventos de segurança, frequentemente registados como *logs*. Contudo, a exploração dos *logs* não é uma tarefa trivial devido à existência de formatos de representação associados a eventos de segurança da informação muito heterógenos.

O contributo que os eventos dão é extremamente importante para colmatar as lacunas existentes nas tecnologias de segurança atuais. Perante estas circunstâncias a principal questão na qual se pretende obter resposta no decorrer desta dissertação, está relacionada com o formato de representação de dados adequado, no âmbito de uma gestão de eventos de segurança de informação integrada.

1.1 Objetivos

A análise de *logs* tem vindo a assumir uma crescente importância na área de segurança da informação. Os desenvolvimentos nesta área específica surgem como uma alternativa viável à necessidade da proteção da informação que se encontra exposta aos mais variados ataques. Os profissionais da área de segurança de informação necessitam cada vez mais de informação e mecanismos de segurança capazes de os auxiliar e orientar de forma inteligente no combate às ameaças aos seus SI. As técnicas de gestão de *logs* têm crescido muito em termos de capacidade e o valor das aplicações *log* têm sido reconhecidos pelas organizações e pela comunidade científica, como sendo uma ferramenta valiosa na deteção de eventos e monitorização do desempenho da rede.

Objetivamente existe uma forte motivação em contribuir com uma investigação que tem como principal foco os formatos de representação de eventos de segurança. Os objetivos e os resultados esperados no âmbito desta dissertação são:

Objetivos esperados:

- Identificar formatos de representação existentes;
- Identificar contextos de utilização desses formatos;
- Avaliar a possibilidade de integrar os diferentes formatos de representação existentes;
- Avaliar a eficácia dos formatos identificados para cada contexto de utilização;
- Explorar os diferentes formatos existentes.

Resultados esperados:

- Definir um formato de representação genérico de eventos de segurança da informação. Este formato deverá integrar a informação proveniente de diversos tipos de sensores, assim como facilitar a respetiva correlação;
- Uma *interface* de eventos de segurança de informação, capaz de transformar os dados obtidos a partir de diversos sistemas de registos *logs* em informação útil, expresso no formato de representação de eventos proposto.

1.2 Método de Investigação

Para realizar esta investigação, a abordagem *Design Science* foi a mais adequada por se tratar do desenvolvimento de um artefacto, neste caso um modelo de representação único de *logs*. Como se pode constatar na Figura 1, o método *Design Science* é composto por cinco fases principais: sensibilização para o problema, sugestão, desenvolvimento, avaliação e conclusão (A. R. Hevner, S. T. March, J. Park, 2004).

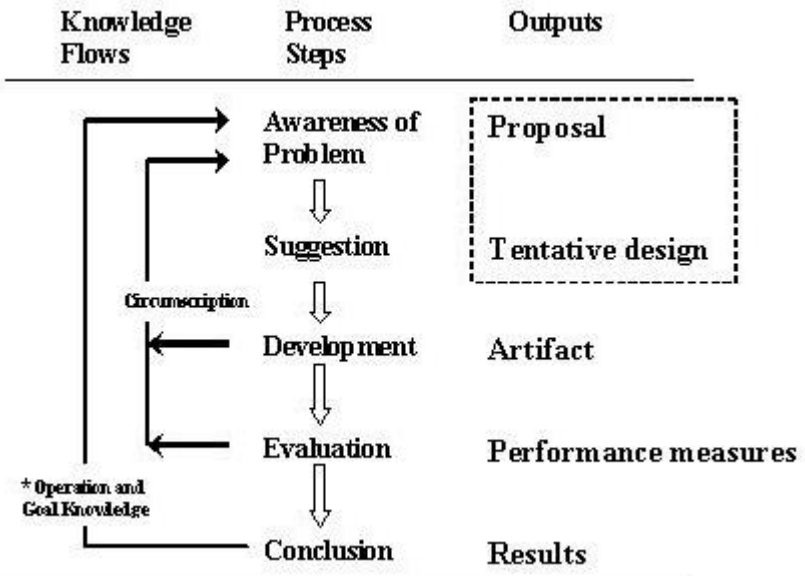


Figura 1 - Metodologia *Design Science* (Fonte: Vaishnavi, V., & Kuechler, (2004))

A primeira fase do processo reflete a sensibilização do problema onde está presente a lacuna de conhecimento, e visa aumentar a familiarização sobre o problema. O resultado desta fase é a descrição do tópico alvo de investigação. O passo que se segue diz respeito à sugestão da solução. Depois de aprofundados os conhecimentos na área, o resultado obtido aquando a finalização da revisão da literatura permite ao investigador ultrapassar as lacunas de conhecimento existentes na fase inicial, servindo de suporte teórico para a investigação. A fase seguinte é o desenvolvimento da solução e é onde está centrada a maior parte do trabalho de investigação, tendo como resultado final o artefacto devidamente definido. A fase da avaliação diz respeito à validação do artefacto. Por fim, a conclusão é a ultima fase e é onde termina o ciclo de conceção do artefacto, determinando a qualidade e impacto do artefacto produzido.

Esta abordagem foi realizada focando as sete orientações estabelecidas por Hevner et.al sendo elas: o artefacto, relevância do problema, rigor na investigação, *design*

como um processo de investigação, avaliação do *design*, contribuições da investigação e comunicação da investigação.

1.2.1 Artefacto

Criação de um formato de representação de eventos de segurança e uma *interface* capaz de recolher eventos de segurança da informação, e transformar numa linguagem comum os eventos obtidos a partir de diversos sistemas de registos (*logs*).

1.2.2 Relevância do Problema

A quantidade, diversidade e heterogeneidade dos eventos de segurança constitui um problema para todos aqueles que querem analisar e tirar proveito do valor dos *logs*.

1.2.3 Rigor na investigação

Todos os resultados, métodos, *software e scripts* desenvolvidos são analisados e revistos mais do que uma vez, de modo a aumentar credibilidade e valor nas conclusões retiradas. Todos os desenvolvimentos são documentados e reportados por forma a permitirem a sua repetição por terceiros.

1.2.4 Processo de pesquisa e avaliação do desenho, contribuições e comunicação da investigação

Criação de uma *interface* e um formato de representação de eventos que possa ser utilizado futuramente pela comunidade científica. A *interface* bem como o formato de representação de eventos serão alvos de validação. O resultado desta dissertação vai estar disponível na biblioteca da Universidade do Minho.

1.3 Organização do Documento e Notas Relevantes

Além da Introdução, onde estão definidos os objetivos, resultados esperados e o método de investigação utilizado, o restante documento é composto por seis secções adicionais: Fundamentos Teóricos de Segurança da Informação, *Log* de Eventos, Análise dos Eventos, Especificação e Implementação do Formato de Representação de Eventos, Conclusões e Trabalho Futuro e Referências Bibliográficas, respetivamente.

A secção dos Fundamentos Teóricos de Segurança da Informação e *Log* de Eventos é constituída pelo enquadramento teórico onde são focados e referenciados aspetos considerados relevantes para a contextualização do tema em si. Sempre que necessário, os vários conceitos são alvos de um nível de detalhe acrescido. No decorrer desta secção o nível de detalhe aumenta, estando enquadrado com o problema inicialmente identificado.

Na secção Análise dos Eventos é apresentada uma contextualização e uma análise detalhada em torno dos eventos *log*. A secção Especificação e Implementação do Formato de Representação de Eventos, como o próprio título sugere, consiste na especificação e implementação do formato de representação proposto. É proposto e especificado um formato e uma *interface* de representação dos eventos.

De seguida, na secção Conclusões e Trabalho Futuro são apresentadas as conclusões. É efetuada uma discussão e uma análise critica sobre todo o trabalho desenvolvido no âmbito da dissertação. Nesta secção contém também as indicações futuras, tendo como base o trabalho desenvolvido.

2. Fundamentos Teóricos de Segurança da Informação

Tanto no contexto operacional como na literatura científica, a Segurança da Informação é um dos tópicos mais abordados e desafiadores do momento, e a questão em torno das potencialidades dos eventos *logs* tem suscitado a curiosidade de muitas organizações e investigadores, que vêm com bons olhos, os l*ogs* de eventos como uma importante técnica e/ou ferramenta no auxilio da segurança da informação.

Perante estas circunstâncias, e de acordo com o tema proposto nesta dissertação é realizada uma revisão sistemática da literatura, de modo a aprofundar os conhecimentos na área e acompanhar a evolução e a preocupação desta temática ao longo do tempo.

2.1 Estratégia de Pesquisa

A estratégia de pesquisa, atendendo que o objetivo inicial passa pela absorção de informação e conhecimento, divide-se em três partes essenciais. A primeira, visto que não existem desenvolvimentos relevantes nesta área e estar fora do âmbito de conhecimento, é a pesquisa e investigação de forma abrangente em torno da área de segurança da informação (embora sempre com o foco nos eventos de segurança) de modo a adquirir conhecimentos sobre o tema em questão. A segunda e terceira parte têm como foco de pesquisa, absorver o que tem sido desenvolvido na área até ao momento, acompanhando a evolução do tema e encontrar conceitos que fundamentem de forma credível as respostas que se pretende obter no decorrer do projeto de investigação.

Face ao enorme volume de documentos científicos digitais, existe uma variedade de serviços e base de dados científicas, que visam suportar a divulgação e pesquisa do trabalho científico. No entanto, diferentes bases de dados científicas têm diferentes filosofias, e isso influencia substancialmente os resultados retornados das pesquisas.

As pesquisas são um processo complexo pois envolvem um número elevado de documentos e ao mesmo tempo um processo dinâmico, na medida em que é necessário

adaptar-se e de certo modo encontrar mecanismos mediante os resultados obtidos nas pesquisas. Para o efeito, os critérios utilizados na pesquisa foram principalmente as datas, citações, autores e a escolha da área do tema em questão.

A estratégia delineada para a seleção da bibliografia consiste em uma recolha inicial de uma série de documentos com critérios bem definidos (mais citados, conferências, revistas cientificas, palavras-chave, entre outros) de forma a no final da pesquisa ter entre 10 a 20 documentos que permitam a escrita e consolidação dos conceitos teóricos.

Em relação às palavras-chaves utilizadas para a pesquisa de documentos que sirvam de base na consolidação dos fundamentos teóricos são apresentadas na Tabela 1.

Tabela 1 – Palavras-chave para a pesquisa no âmbito da dissertação

Palavras-Chave	***Keywords***
Eventos de segurança	*Security Event*
Correlação eventos *log*	*Log Event Correlation*
Gestão de Segurança de Informação	*Security Information Management*
Gestão de Segurança de Eventos	*Security Event Management*
Gestão de eventos	*Log Management*
Evento	*Event*

Após o levantamento dos documentos, é necessário selecionar os documentos mais relevantes, aplicando critérios rigorosos. Um dos critérios de seleção foi a contabilização do número de citações, pois é um indicador relevante para avaliar a importância de uma produção científica. Porém, a sua interpretação tem de ser cuidada, pois publicações recentes têm citações reduzidas. O autor, título e o resumo também são cruciais na seleção, principalmente o resumo pois é o intermediário entre o título e o artigo e contém informações importantes para o tema alvo de investigação. As palavras-chave bem documentadas também são alvo de atenção, pois refletem os aspetos mais importantes do documento em questão.

2.2 Conceitos Teóricos

Nesta secção são apresentados conceitos considerados basilares, tendo como principal objetivo clarificar aspetos considerados essenciais no âmbito do tema desta dissertação.

2.2.1 Segurança da Informação

É um truísmo dizer que a informação é a moeda da era da informação e em muitos casos, a informação é um dos ativos mais valiosos que as organizações possuem (Calder, 2006). Contudo, o valor da informação advém das características que possui e quando essas características se alteram, o valor da informação tanto aumenta, ou mais *comummente*, diminuí (Whitman & Mattord, 2009). Assim a segurança de informação é extremamente importante para garantir que esses recursos são bem protegidos (Susanto, Almunawar, & Tuan, 2011). A segurança de informação, segundo a ISO 27001:2005 é a preservação da confidencialidade, integridade e disponibilidade (CIA) da informação. Em alguns casos, outras propriedades como autenticidade, responsabilidade e confiança podem estar envolvidas.

2.2.2 Incidentes de Segurança

Existe pouco consenso sobre a definição das palavras incidente, ataques e eventos. No domínio dos incidentes de segurança, Howard e Longstaff desenvolveram esforços para estrutruturar numa linguagem comum, uma taxonomia de incidentes como ilustra a Figura 2. Nessa taxonomia, são representadas as relações de eventos para ataques e para incidentes. Com esta definição pretende-se clarificar e distinguir os conceitos de incidente, ataque e evento.

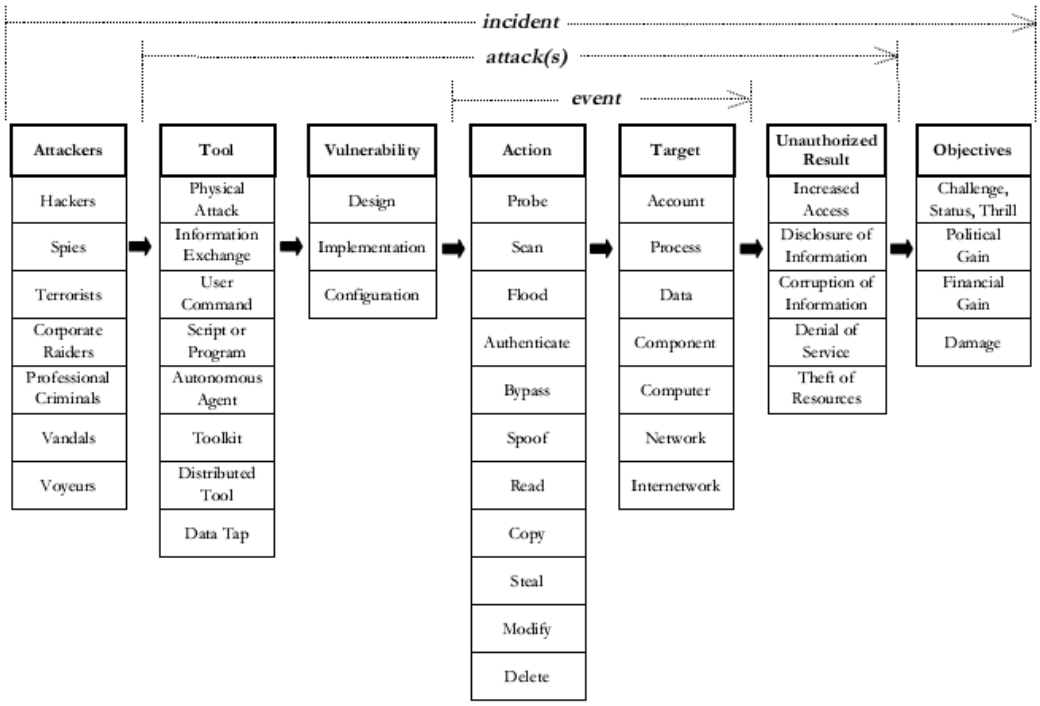


Figura 2 - Incidentes de Segurança (Fonte: Howard & Longstaff (1998))

Um incidente é qualquer ataque claramente identificado sobre os ativos da organização (Whitman & Mattord, 2009). Os autores Howard e Longstaff classificam um incidente como um grupo de ataques que pode ser distinguido de outros ataques devido à especificidade dos atacantes, dos ataques, dos objetivos, dos locais e do tempo.

2.2.3 Ataques

Segundo Whitman e Mattord, um ataque é considerado como um ato que se aproveita de uma vulnerabilidade de forma a comprometer um sistema controlado. É consumado por um agente de ameaça que prejudica ou rouba informações de organizações ou ativos físicos. A vulnerabilidade é uma fraqueza identificada no sistema controlado, onde os sistemas de controlo não estão presentes ou não são efetivamente suficientes. Ao contrário das ameaças, que estão sempre presentes, ataques só existem quando um ato específico pode causar perdas ou danos (Whitman & Mattord, 2009). De seguida são apresentados e descritos alguns dos principais tipos de ataques de segurança:

- *Denial-of-Service and Distributed Denial-of-Service*

Os ataques DoS são provavelmente os mais desagradáveis e os mais complicados de resolver, pois são fáceis de lançar, difíceis de controlar e não é fácil recusar os pedidos do atacante sem também recusar os pedidos dos serviços legítimos. Nos ataques DoS, os atacantes enviam um enorme número de conexões ou pedido de informações a um alvo, que origina uma sobrecarga do sistema e impossibilita o alvo de responder a pedidos legítimos do serviço (Whitman & Mattord, 2009) . Os ataques DoS envolvem ferramentas de tecnologias que geram e enviam pacotes a partir de uma única fonte, destinada a um único destino (Houle, Weaver, Long, & Thomas, 2001).

Nos ataques DoS distribuídos vários sistemas são comprometidos de forma a lançar um ataque DoS para um alvo específico. As máquinas comprometidas são transformadas em *zombies*, máquinas essas que são direcionadas remotamente pelo atacante para participar no ataque (Whitman & Mattord, 2009).

- *Spoofing*

Spoofing é uma técnica utilizada para ocultar a proveniência, onde o intruso envia mensagens com um endereço de *Internet Protocol* (IP) de origem falsificado, de forma a indicar que as mensagens estão a vir de um *host* confiável (Whitman & Mattord, 2009). As técnicas de *spoofing* podem ser categorizadas em diferentes tipos, de acordo com o endereço de origem falsificado utilizado nos pacotes. (Chen & Yeung, 2006). Os três tipos mais comuns de falsificação IP são a falsificação aleatória, falsificação de sub-rede e falsificação fixa (Mirkovic & Reiher, 2004).

- *Man-in-the-Middle*

Os dados transmitidos em qualquer tipo de rede estão sujeitos a serem intercetados sem a devida autorização. Um dos mais difundidos métodos de ataque, a uma máquina que recorre à interceção de dados, é o *man-in-the-middle* ou *Transmission Control Protocol* (TCP) *hijacking* como também é designado. Este tipo de ataque utiliza o IP *spoofing* para permitir que o atacante se assuma como outra entidade na rede. Uma variante do TCP *hijacking*, envolve a interceção de uma troca de chaves de criptografia, que permite ao *hacker* agir como um invisível *man-in-the-middle* (Whitman & Mattord, 2009).

- *Snnifers*

O ataque *Sniffer* é uma atividade que resulta de um utilizador malicioso que tenta obter informações sobre a rede ou tráfego dessa rede. A maior parte das vezes é um programa de captura de pacotes que duplica os conteúdos dos pacotes que circulam pela rede. Também é conhecido pelo ataque nas conexões iniciais entre cliente e servidor, de forma a obter as credenciais de acesso. Quando realizado corretamente os ataques por meio *Snnifer* são invisíveis para todas as entidades da rede e frequentemente precedem aos ataques por *spoofing* ou TCP *hijack* (Stewart, Chapple, & Gibson, 2012).

2.2.4 Eventos

As operações de computadores e redes envolvem inúmeros eventos. Uma organização que se tem dedicado a abordar a temática dos eventos é a MITRE[1]. Segundo eles, um evento é uma ocorrência dentro de um ambiente, que envolve geralmente uma tentativa de mudança de estado. Esta mudança de estado inclui geralmente a noção do tempo, a ocorrência e quaisquer detalhes que explicitamente dizem respeito ao acontecimento ou ambiente que pode ajudar a explicar ou compreender as causas do evento ou efeitos.

Segundo Howard e Longstaff, do ponto de vista da segurança do computador e da rede, as mudanças de estado resultam de ações que são dirigidas contra alvos específicos. Na Figura 2, desenvolvida pelos mesmos autores e baseada nas suas experiências, é apresentada também uma matriz de ações e alvos possíveis entre os eventos de computadores e rede. Os autores defendem e dão ênfase que para a ocorrência de um evento acontecer, deve existir uma ação realizada e tem de ser dirigida contra um alvo, contudo a ação não tem que ter realmente sucesso na mudança de estado do alvo.

2.2.5 Mecanismos de controlo de segurança

Para a defesa de vários ataques e com o objetivo de melhorar a proteção dos ativos das organizações, diversos sistemas de segurança são implementados, como por exemplo os antivírus, rede privada virtual, *firewall*, entre outros. De forma a compreender melhor os mecanismos e/ou componentes de segurança, alguns são apresentados nos pontos seguintes:

- *Firewall*

A *Firewall* é um sistema de defesa que separa a rede local e a rede externa. É considerado como um grupo de medidas de prevenção. Embora todas as *firewalls* desempenhem a função de *logs*, cada fabricante utiliza uma gestão e estratégia

[1] CEE – The MITRE Corporation

diferente o que origina diferenças nos formatos *log*, tais como *Syslog*, *Traffic log*, *WELF, FortiGaet 3000*, entre outros (Zhaojun, Yong, & Wenjing, 2010).

- *Intrusion Detection System*

Intrusion Detection Systems (IDS) é um tipo de controlo de segurança de computadores, cujo objetivo é monitorizar a atividade de um SI para detetar a ocorrência de atividades maliciosas (Morin, Mé, Debar, & Ducassé, 2009). É um sistema que recolhe informações a partir de uma variedade de sistemas (computadores ou equipamento de redes) e posteriormente analisa sinais de ataque e utilização indevida. Os principais problemas com os IDS prendem-se ao facto de gerarem diariamente um número elevado de alertas, um elevado número de falsos positivos que são misturados com alertas verdadeiros. Atualmente os IDS detetam ataques de baixo nível e não conseguem detetar todos os ataques (Amiri, Gharaee, & Enayati, 2011).

- *Intrusion Prevention System*

Intrusion Prevention System (IPS) são dispositivos de segurança de rede que monitorizam atividades maliciosas na rede. As principais funções dos IPS são as de identificação de atividades maliciosas, *logs* de informações sobre atividades, tentativa de parar ou bloquear atividades, e relatório de atividades (Scarfone & Mell, 2007). Um problema real na utilização dos IPS é a grande produção de falsos positivos e principalmente a sua capacidade de apenas gerir ataques já conhecidos. Na verdade, este tipo de sistemas não pode utilizar produtivamente a experiência passada na deteção (ou não) de utilização indevida (Colace, De Santo, & Ferrandino, 2012).

2.2.6 Tecnologias de segurança

Na subsecção 2.2.5 foram apresentados alguns mecanismos de controlo de segurança, controlo esses, essenciais para um bom planeamento da segurança da informação (Whitman & Mattord, 2009). Na Tabela 2 são enunciadas algumas tecnologias de segurança existentes no mercado, responsáveis pela produção dos mais variados eventos de segurança.

Tabela 2 – Exemplo de aplicações relacionadas com controlo de segurança

	Mecanismos de controlo	Tecnologias de segurança
Eventos de Segurança	**NIDS**[2]	Cisco CSA, Cisco IDS, Enterasys Dragon, Fortinet Fortigate, Juniper ISG, SNORT, Niksun NetVCR, SourceFire Intrusion Sensor
	IPS	ForeScout CenterACT, Juniper NetScreen IDP, McAfee Intrushield, Radware Defense Pro, FireEye, Tipping Point X, IPAngel
	Firewalls	CheckPoint, Linux Iptables, PaloAlto PA, Cisco ACE
	VPN[3]	ArraySP, Nortel VPN Gateway, Checkpoint VPN-1, Cisco ASA
	Antivírus	McAfee, Sophos, Symantec, Trend Micro
	HIDS[4]	OSSEC
	Vulnerability Scanner	Nessus

2.2.7 Security Information and Event Management

De modo a lidar com os dados produzidos em enormes quantidades pelos *logs* de segurança e proporcionar um maior nível da segurança da informação, os *Security Information and Event Management* (SIEM) surgem como solução viável. Os SIEM auxiliam também as organizações que lutam constantemente com as diversas regulamentações de conformidade existentes.

O acrônimo SIEM é atribuído a analistas da empresa *Gartner Group* e deriva de duas distintas, mas complementares, tecnologias: *Security Event Management* (SEM) e *Security Information Management* (SIM). Durante a última década, estas duas tecnologias têm convergido em uma única solução, conhecida atualmente como SIEM (Hernandez, 2010).

SIEM são tipicamente aplicações empresariais que assentam em cima de todos os dispositivos de segurança de uma rede. Isto inclui, mas não está limitado a, *firewalls*, *IDS*, scanners de vulnerabilidade, base de dados e aplicações. Todos estes sistemas

[2] *Network-Based Intrusion detection system*
[3] *Virtual Private Network*
[4] *Host-based intrusion detection system*

alimentam o sistema SIEM, o qual pode utilizar algoritmos estáticos para analisar dados dos *logs* ou permitir que se possa criar regras personalizadas que procurem padrões específicos em dados de *logs* (Chuvakin, Schmidt, & Phillips, 2012). Os SIEM também podem auxiliar uma organização no cumprimento dos regulamentos relativos à retenção de dados e este último pode ser útil em casos de *e-discovery* e análise forense. Outros casos de utilização incluem a gestão de utilizadores e monitorização de políticas e gestão de identidades (Karlzén, 2009).

2.2.7.1 *Anatomia de um SIEM*

Um SIEM pode ser comparado a uma máquina complexa na medida em que um SIEM tem várias partes móveis, cada um realizando uma tarefa específica. Existem variações sobre o SIEM padrão, com partes específicas adicionais, mas um SIEM simples pode ser dividido em partes ou processos separados. Cada uma dessas partes pode trabalhar de forma independente dos outros, mas sem eles todos a trabalhar em conjunto, o SIEM como um todo não funcionará corretamente (Miller & Pearson, 2011).

Naturalmente que existem diferenças entre as variadas soluções SIEM, contudo existem certos conceitos que são inerentes a todos eles. Esses conceitos fundamentais estão ilustrados na Figura 3.

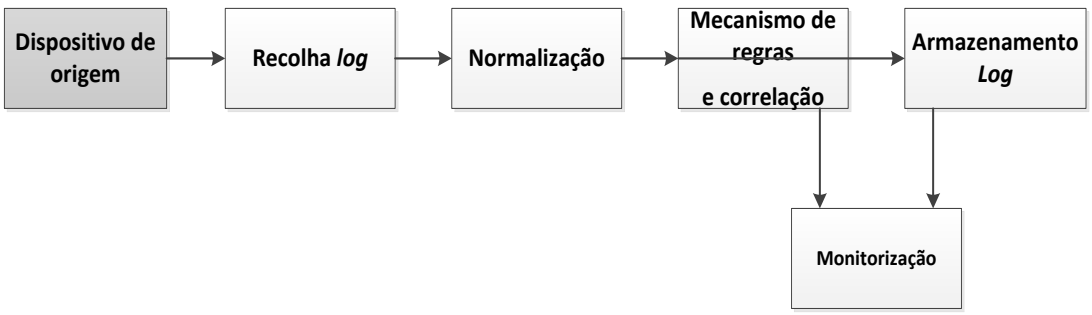


Figura 3 - Anatomia do SIEM (Adaptado: Miller & Pearson, (2011))

De seguida é apresentada a descrição de cada uma das componentes presentes no SIEM:

- Dispositivo de origem

A primeira parte de um SIEM são os dispositivos de origem que alimentam o SIEM com informação. Um dispositivo de origem é o dispositivo em que se pretende obter informação, para posteriormente armazenar e processar (Miller & Pearson, 2011). O dispositivo de origem não é uma componente efetiva do SIEM, quando se olha para o SIEM como uma aplicação, contudo é uma componente vital de todo o processo, na medida em que todos os sistemas da rede estão a processar algum tipo de informação. Segundo o mesmo autor, sem o dispositivo de origem e sem a informação que esses dispositivos geram, o SIEM é apenas uma boa aplicação que não faz nada.

- Recolha *log*

O passo seguinte no fluxo é encontrar alguma forma de obter os diferentes *logs*. Os SIEM recolhem dados de *logs* de uma enorme quantidade de diferentes tipos de dispositivos – incluindo aplicações, sistemas operativos, *firewall, router & switches,* IDS e IPS, servidores *proxy*, sistemas de controlo de acesso, entre outros – que geram dados. Os dados são o artefacto que exprime o que está a acontecer atualmente, e pode ser um evento ou apenas informações de configuração ou de estado do dispositivo (Jones, 2010).

A recolha de dados ocorre num número variado de formas dependendo da solução e do sistema final. Segundo Miller & Pearson, o processo de recolha de *logs*, na sua forma mais básica, pode ser dividido em dois métodos fundamentais de recolha: ou o dispositivo de origem envia os seus *logs* para o SIEM, que é chamado de método *push,* ou o SIEM alcança e recupera os *logs* do dispositivo de origem e é chamado de método *pull.*

Os protocolos é a forma como os coletores comunicam com a origem, existindo diversas modos de transmissão. Um método de transmissão de eventos pode ser através de protocolos de rede tais como o *Simple Network Management Protocol* (SNMP), *Netflow* ou *Internet Protocol Flow Information Export* (IPFIX). Em outras circunstâncias, os dispositivos de origem geram os eventos num formato comum,

como por exemplo o *Syslog*, *Windows Event Log* ou *Syslog-NG*, que será lido posteriormente pelo mecanismo responsável de recolha dos *logs*. Para fontes de *logs* que não suportem esses protocolos, ou outros protocolos *standard*, pode ser utilizado um agente. Um agente é uma parte de um *software*, instalado na origem dos *logs*, que traduz (normaliza) os dados dos *logs* para um formato que o SIEM entenda (Karlzén, 2009). A utilização de agentes significa geralmente maior tempo de implementação para o produto, embora possa ser também utilizado para a normalização e outro pré processamento necessário (Murray, 2003).

- Normalização

Um dos maiores obstáculos quando estamos a lidar com diferentes dispositivos de segurança é que cada *log* tem um diferente formato. Com tantos e diferentes tipos de formato, a maior parte dos SIEM normalizam os dados para um formato proprietário (Karlzén, 2009). O ato de modificar todos esses diferentes tipos de *logs* em um único formato é chamado de normalização. Cada tipo de SIEM vai lidar com o ato da normalização de diferentes maneiras, mas o resultado final é fazer com que todos os *logs*, não tendo em conta o tipo de dispositivo ou fabricante, tenha a mesma aparência para o SIEM (Miller & Pearson, 2011). Segundo Hernandez o processo de normalização extrai informação comum e expressa-la em um formato consistente, o que permite uma comparação direta de diferentes eventos.

- Mecanismo de regras e correlação

A correlação é a função de ligar vários eventos de segurança ou alertas, normalmente dentro de um determinado intervalo de tempo e em vários sistemas, para identificar atividades anómalas que não seriam evidentes a partir de qualquer evento singular. Para isso ser possível a solução SIEM deve ter regras em vigor que instruam o mecanismo de correlação sobre os tipos de eventos que devem tentar correlacionar e as condições que justifiquem um alerta. Através de Miller & Pearson podemos perceber que um mecanismo de regras expande sobre a normalização dos eventos das diferentes origens, de modo a despoletar alertas dentro do SIEM, devido às condições específicas nesse *log*. O método de escrever regras geralmente inicia-se de uma forma bastante simples, mas pode-se tornar bastante complexa. Segundo os mesmos autores, um mecanismo de correlação é um subconjunto do mecanismo de regras. O que o

mecanismo de correlação faz é combinar múltiplos eventos padrão provenientes das diferentes fontes, em um único evento correlacionado.

- Armazenamento *log*

Ao serem analisados, os dados são geralmente armazenados *online* e quando não são necessários de imediato são armazenados. Os dados podem ser armazenados normalizados de forma a acelerar o processo caso seja necessário novamente a sua utilização (Karlzén, 2009). A forma mais usual por parte dos SIEM para armazenamento de dados são plataformas de bases de dados padrão, ou uma outra aplicação de "grandes base de dados" utilizados empresarialmente. A utilização das bases de dados é uma boa solução para o armazenamento dos *logs*, mas algumas questões podem surgir dependendo de como o SIEM implementa a base de dados (Miller & Pearson, 2011).

Diferentes modelos de armazenamento são utilizados para lidar com as questões de escalabilidade e gestão dos dados. No início, a maior parte dos SIEM e gestão dos *logs,* foram construídas sobre base de dados relacionais para armazenar eventos e registos de *log*s (Jones, 2010). Segundo o mesmo autor, nos modelos atuais a plataforma SIEM mapeia cada atributo dos dados em colunas, de modo a que cada evento seja armazenado em uma única linha da base de dados.

- Monitorização

A etapa final da anatomia de um SIEM é a monitorização e consiste na interação com os *logs* armazenados. Uma vez que os todos os *logs* estão no SIEM e todos os eventos foram processados, é necessário encontrar forma de utilizar inteligentemente essa informação. Para ser possível essa interação os SIEM possuem uma *interface console* que vai ser o principal meio de comunicação com os dados armazenados no SIEM. O SIEM torna a visualização e análise de todos esses *logs* muito mais fácil porque este normaliza os dados (Miller & Pearson, 2011).

2.2.7.2 *Tecnologias e Soluções SIEM*

Segundo o relatório do quadrante mágico para os SIEM da Gartner, o mercado SIEM é definido pela necessidade do cliente analisar dados de eventos de segurança em tempo real para a gestão de ameaças internas e externas, e para recolher, armazenar, analisar e reportar os dados dos *logs* para resposta a incidentes, análise forense e conformidades regulamentares. Os vendedores incluídos na análise do quadrante mágico têm tecnologias projetadas para o mencionado anteriormente, atuam ativamente no mercado e vendem essas tecnologias para centros de segurança.

De seguida, é sucintamente descrita uma solução SIEM presente no quadrante mágico e utilizada no âmbito desta dissertação.

- *Open Source Security Information Management (OSSIM)*

O projeto desenvolvido pela AlienVault é *open source* e portanto é passível de instalar, modificar e experimentar com rigor. Contudo a versão gratuita desta tecnologia tem algumas limitações no que diz respeito ao desempenho de armazenamento. Como Miller e Pearson referem no seu livro, e bem, o conceito do OSSIM é de não reinventar a roda. Na diversidade de ferramentas *open source* disponíveis e com créditos dados, o OSSIM aproveita esse facto e agrega-os numa solução poderosa para as suas operações de segurança.

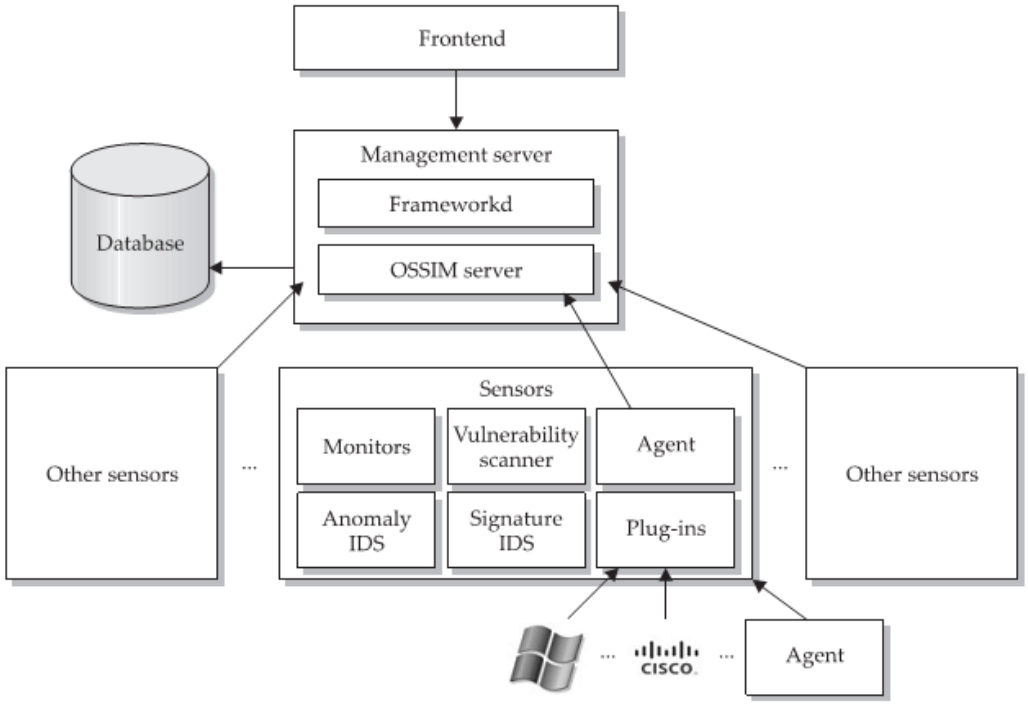


Figura 4 - Modelo OSSIM (Fonte: Miller & Pearson (2011))

Como é possível constatar na Figura 4, o OSSIM tem quatro componentes que funcionam tipicamente como blocos, sendo eles os sensores, o servidor de gestão, a base de dados e o *front-end.*

.

Esta página foi colocada propositadamente em branco.

3. *Log* de Eventos

Nesta secção pretende-se descrever as contribuições efetuadas na área dos eventos, com especial foco nos formatos de representação de eventos de segurança da informação, de forma a avaliar o estado da arte em torno da diversidade dos formatos dos eventos de segurança.

3.1 Breve descrição do Problema

Um dos problemas em torno dos *logs*, além da inconsistência do seu conteúdo, da variedade dos *timestamp* e da sua origem, é a inconsistência dos formatos que o representa. Os *logs* devem ter um formato padronizado de modo a que se possa tirar partido das potencialidades dos *logs* de forma eficiente e inteligente. Todos os tipos de eventos de segurança devem ser normalizados num formato único (Li, 2010). Este mesmo autor destaca a importância para a normalização da representação de cada propriedade do campo, como os endereços de origem e destino, "nome do *host"*, entre outros, e a configuração e sincronização do valor do tempo.

Esta inconsistência dos formatos de *logs* e campos de representação dos dados apresenta desafios para as pessoas que pretendem fazer análise dos *logs* e para quem pretende compreender o significado dos diversos campos de dados em cada *log* (Kent & Souppaya, 2006).

3.2 Caracterização dos *Logs*

A deteção de comportamentos mal-intencionados num SI é um problema amplo e difícil, com ameaças que vão desde *hackers* e *malware* a ataques internos. Um método que se demonstra eficaz para detetar e combater esses comportamentos é a análise das mensagens *logs* (Myers, Grimaila, & Mills, 2011). Contudo, esse método é dispendioso, difícil, e muitas organizações não conseguem obter sucesso (Wilshusen & Powner, 2009). Os ficheiros de *log* são geralmente muito grandes para extrair

informação valiosa, apesar de a técnica manual ser umas das mais utilizadas (Shrivastava, 2012).

Os *logs* são então uma coleção de registos de eventos e que segundo Kent e Souppaya podem conter uma grande variedade de informações sobre os eventos que ocorrem em redes e sistemas. Sob diversas circunstâncias, muito dos *logs* gerados dentro das organizações podem ter alguma relevância para a segurança dos SI. Do ponto de vista dos recursos do sistema, os eventos de segurança consistem em todas as informações relacionadas com a segurança, que são resultado de todos os comportamentos duvidosos que põem em risco a CIA (Jingxin & Zhiying, 2007). Uma análise frequente de *logs* é benéfica para identificar incidentes de segurança, violações de políticas, atividades fraudulentas e problemas operacionais (Kent & Souppaya, 2006). No entanto, defendem os mesmos autores que, para a segurança da informação, esses *logs* normalmente são apenas utilizados como fontes de informação suplementares.

Como referido anteriormente, os *logs* servem muitas funções dentro das organizações. Contudo, as informações de eventos registadas em *logs* não apresentam uma sintaxe e semântica comum. Enquanto as aplicações estão a ser desenvolvidas por pessoas com diferentes formações e aptidões de desenvolvimento, cada aplicação pode gerar a sua própria mensagem e estrutura de *log*, como se pode verificar na Figura 5. Por conseguinte, encontra-se em qualquer SI um conjunto de *logs* distribuídos e heterogéneos que não estão organizados do mesmo modo (Hammoud, 2009). Cada linha de um ficheiro *log* é uma combinação de um tipo de mensagem estática e informações de parâmetros variáveis (Shrivastava, 2012).

Os autores Kent e Souppaya dividem os *logs* em duas categorias: *Logs* de segurança de *software* e de sistemas operativos. Os *logs* de segurança de *software* contêm principalmente informações relacionadas com a segurança de computadores. A maioria das organizações utiliza vários tipos de *software* de segurança baseados em rede e *host* para detetar atividades maliciosas, proteger dados e sistemas e apoiar os esforços de resposta a incidentes. Desta forma o *software* de segurança é uma das principais fontes de dados de *logs* de segurança. Na Figura 5 são apresentados alguns exemplos de *logs* com base nessas fontes de dados.

Intrusion Detection System

```
[**] [1:1407:9] SNMP trap udp [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/06-8:14:09.082119 192.168.1.167:1052 -> 172.30.128.27:162
UDP TTL:118 TOS:0x0 ID:29101 IpLen:20 DgmLen:87
```

Personal Firewall

```
3/6/2006 8:14:07 AM,"Rule ""Block Windows File Sharing"" blocked (192.168.1.54,
netbios-ssn(139))","Rule ""Block Windows File Sharing"" blocked (192.168.1.54,
netbios-ssn(139)).  Inbound TCP connection.  Local address,service is
(KENT(172.30.128.27),netbios-ssn(139)).  Remote address,service is
(192.168.1.54,39922).  Process name is ""System""."

3/3/2006 9:04:04 AM,Firewall configuration updated: 398 rules.,Firewall configuration
updated: 398 rules.
```

Antivirus Software, Log 1

```
3/4/2006 9:33:50 AM,Definition File Download,KENT,userk,Definition downloader
3/4/2006 9:33:09 AM,AntiVirus Startup,KENT,userk,System
3/3/2006 3:56:46 PM,AntiVirus Shutdown,KENT,userk,System
```

Antivirus Software, Log 2

```
240203071234,16,3,7,KENT,userk,,,,,,,16777216,"Virus definitions are
current.",0,,0,,,,,0,,,,,,,,,,SAVPROD,{ xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx },End
User,(IP)-192.168.1.121,,GROUP,0:0:0:0:0:0:0,9.0.0.338,,,,,,,,,,,,,,,
```

Antispyware Software

```
DSO Exploit: Data source object exploit (Registry change, nothing done)  HKEY_USERS\S-
1-5-19\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0\1004!=W=3
```

Figura 5 - Exemplo tipo de *logs* de segurança de *software* (Fonte: Kent & Souppaya, (2006))

Os *logs* de sistemas operativos contêm uma grande variedade de informação e dizem respeito aos sistemas operativos de servidores, *workstations* e dispositivos de rede (routers, *switches*, entre outros). A Figura 6 ilustra o *log* de um evento de um sistema operativo.

```
Event Type:  Success Audit
Event Source:Security
Event Category:     (1)
Event ID:    517
Date:        3/6/2006
Time:        2:56:40 PM
User:        NT AUTHORITY\SYSTEM
Computer:    KENT
Description:
The audit log was cleared
Primary User Name: SYSTEM      Primary Domain: NT AUTHORITY
Primary Logon ID: (0x0,0x3F7)  Client User Name: userk
Client Domain: KENT            Client Logon ID: (0x0,0x28BFD)
```

Figura 6 - Exemplo de tipos *log* de sistema operativo (Fonte: Kent & Souppaya, (2006))

3.3 Formatos de *Log* de Eventos e Protocolos de transmissão

Como tem sido evidenciado no decorrer desta dissertação, existe uma diversidade enorme de formatos para a representação de *logs* de eventos, bem como diferentes protocolos de transmissão. Os formatos de *logs* variam de solução para solução, e além dos formatos proprietários é comum encontrar formatos de texto binários, em *Comma Separated Values* (CSV), *Extensible Markup Language* (XML), Syslog, *JavaScript Object Notation* (JSON), base de dados, entre outros. Sucintamente, são explicados de seguida alguns dos formatos:

- *Syslog*

O protocolo *syslog* foi introduzido pelo *Computer Science Research Group* (CSRG) da Universidade de California-Berkeley como parte da distribuição de *software* Berkeley. O *syslog* foi desenvolvido para fornecer a capacidade de reportar os eventos do sistema. Estes eventos são recolhidos por um processo e registados num ficheiro *log* de eventos local, remoto ou em ambos (Nawyn, 2003). O protocolo define 24 *Syslog Message Facilities* e 8 *Syslog Message Severities* (Gerhards, 2009).

- *Extensible Markup Language*

O XML sendo uma metalinguagem – uma linguagem para descrever outras linguagens - é muitas vezes escolhida para representar o formato intermediário desejado. O XML é uma versão simplificada da *Standard Generalized Markup Language* (SGML), uma sintaxe para marcação de texto especificado e definido pela norma ISO 8879. O XML oferece vantagens, como a flexibilidade, possibilidade de partilha de informações e transferência de *logs* entre as diferentes aplicações e/ou sistemas sem qualquer necessidade do sistema *kernel*, pois armazena diretamente os dados em ficheiros, em base de dados, ou diretamente em outras aplicações.

- *JavaScript Object Notation*

JSON é um formato de texto para a serialização de dados estruturados e utiliza convenções familiares para quem desenvolve, como é o caso da linguagem da família "C", *Java*, *JavaScript*, *Perl*, *Python*, entre outros. Contudo continua a ser uma linguagem e plataforma independente. A simplicidade do formato facilita a interpretação e análise tanto para os seres humanos, como para os computadores.

- *Protocol Buffers*

O *Protocol Buffers* é um eficiente e flexível mecanismo automatizado para a serialização de dados estruturados, muito semelhante ao XML, sendo mais fácil, pequeno e rápido (Google, 2012). Ainda pela mesma entidade, é possível definir como os dados vão ser estruturados e uma vez definido é possível utilizar o código fonte gerado para leitura ou escrita dos dados estruturados para uma variedade de *data streams* e utilizar uma variedade de linguagens de programação.

3.4 Normalização dos formatos *log*

As normas providenciam um guião que permite a construção de soluções que supostamente aumentará a eficiência na organização (ISO/IEC, 2009). A ISO/IEC e o *National Institute of Standards and Technology* (NIST) são entidades que, a nível mundial, operam na criação de normas e também fornecem recomendações para uma gestão eficaz e eficiente dos *logs*. A título de exemplo, a NIST na sua publicação especial NIST-800-92 efetua recomendações que afetam a gestão de *logs,* incluindo o estabelecimento de políticas e procedimentos, a priorização da gestão de *logs*, criação e manutenção de uma infraestrutura de gestão de *logs*. Contudo, apesar de estas recomendações serem importantes, são específicas para atuar a um nível mais alto da gestão de *logs*, não havendo até ao momento uma norma específica que sirva como um guião para a uniformização dos formatos de *log*.

Os formatos de representação de eventos de segurança de distintas fontes de segurança são também diferentes (GU & Li, 2011). Existem algumas propostas para uniformizar os formatos de *log*, como o formato W3C e SNMP. Contudo, a maioria das aplicações utilizam formatos *ad hoc* de *logs* não uniformizados. O grande volume de *logs* gerado

pelas aplicações limita a utilização de técnicas de análise manual, sendo necessário técnicas automatizadas para processamento de *logs*. No entanto, a automatização dos *logs* é uma técnica complexa devido às estruturas pouco definidas e a um enorme vocabulário de termos (Jiang & Hassan, 2008). O desenvolvimento de um formato padrão permitirá assegurar a interoperabilidade entre os demais recursos (Fooprateepsiri & Kurutach, 2010).

Existem várias tentativas anteriores para desenvolver *standards* de interoperabilidade nos *logs* e eventos, em especial no domínio da segurança de computadores. Mas, por uma razão ou outra, não conseguiram ter o impacto ou a importância esperada. Alguns dos esforços mais notáveis e que contribuíram, direta ou indiretamente, para normalização dos eventos ou *log* (mesmo que num âmbito mais limitado) são destacados de seguida.

- *Common Intrusion Detection Framework*

O *Common Intrusion Detection Framework* (CIDF) que incluiu a definição de uma linguagem designada por *Common Intrusion Specification Language* (CISL) foi patrocinado pela *Defense Advanced Research Projects Agency* (DARPA). O CIFD tinha como objetivo a definição de uma linguagem capaz de permitir que os sistemas IDS pudessem expressar e compartilhar dados de intrusão relevantes (Kahn, Porras, Staniford-Chen, & Tung, 1998). Esta iniciativa posteriormente fundiu os seus esforços com uma outra, *Intrusion Detection Message Exchange Format.*

- *Intrusion Detection Message Exchange Format*

O *Intrusion Detection Message Exchange Format* (IDMEF) é um dos mais representativos formatos, para deteção de intrusões, reconhecido internacionalmente e é aplicado em muitas organizações. Com este formato é possível simplificar a análise de eventos de segurança, evitar a complexidade do conteúdo do evento e também facilitar a correlação de eventos de segurança (GU & Li, 2011). O objetivo do IDMEF é definir formatos de dados comuns e procedimentos de intercâmbio de partilha de informações de interesse para deteção de intrusão e sistemas de resposta (Cuppens, 2001). É baseado num modelo de dados orientado a objetos, e adota a linguagem *XML*

para realizar uma descrição formalizada, sendo consistente compatível, e atende às necessidades de padronização de evento de segurança em ambientes heterogêneos.

- *Incident Object Description Exchange Format*

O objetivo do *Incident Object Description Exchange Format* (IODEF) é o da definição de um formato de dados comum para descrever e trocar informações de incidentes, normalmente com uma entidade central designada por *Computer Security Incident Response Teams* (CSIRTs) (Danyliw, Meijer, & Demchenko, 2007). Tal como os eventos, os incidentes são também inerentemente heterogéneos, a sua informação advém de diferentes origens e contém informação sensível. Apesar de atuar em campos diferentes, deve ser visto como um complemento à normalização dos eventos.

- *Common Event Expression*

A especificação *Common Event Expression* (CEE) desenvolvida pela corporação MITRE pretende resolver o problema da normalização da representação dos eventos, fornecendo ferramentas e agrupando-as em torno de quatro áreas: dicionários de terminologias, representação, transporte e recomendações. Como é possível visualizar na Figura 7, essas áreas são mapeadas diretamente com as quatro componentes da arquitetura CEE: *common dictionary events taxonomy* (CDET), c*ommon log sintax* (CLS), c*ommon log transport* (CLT), c*ommon events log recomendations* (CELR).

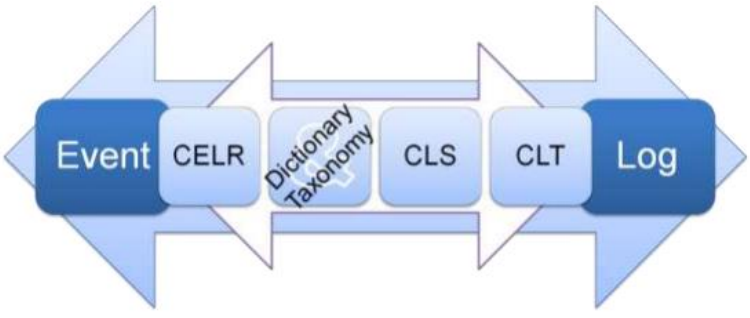


Figura 7 - Componentes da Arquitetura CEE (Fonte: MITRE, (2011))

O dicionário CEE define a terminologia do evento (como por exemplo, o nome dos campos), a taxonomia sugere entradas de como classificar os eventos e o CLS define como o evento é representado. O CLT é o responsável pelo transporte das informações dos eventos e, por último, o CELR fornece orientações sobre como os eventos e campos relacionados devem ser registados.

No que diz respeito aos formatos de representação de eventos, a componente CLS é a responsável pela forma como os dados dos eventos são representados. Além de definir o formato geral de representação de eventos, este componente define um certo número de maneiras diferentes para codificar o evento e os campos do evento. Uma vez que cada codificação é baseada na mesma estrutura do evento, a tradução entre diferentes codificações CLS é eficiente e simples. Com base nas informações da comunidade CEE, CLS apoia minimamente o XML, JSON, e Syslog. São tidas também considerações em como fornecer compatibilidade com outras codificações, como sintaxes binárias e o formato *W3C Extended Log Format*.

- *Common Event Format*

Outro contributo para a normalização da sintaxe dos eventos é o *Common Event Format* (CEF) da Hewlett-Packard, que face à integração complexa que advém da infinidade de formatos provenientes de dispositivos diferentes, contribui com a interoperabilidade entre os vários dispositivos de eventos ou produtores de *logs*. Alinha os *logs* de saídas dos dispositivos com um formato comum, contendo estes a informação mais relevante dos eventos. O CEF pode ser facilmente adotado por fornecedores de dispositivos de segurança e não só. Este formato contém a informação mais relevante do evento de forma a tornar a sua análise e utilização mais fácil (ArcSight, 2010).

- *Common Base Event*

Common Base Event (CBE), proposto pela IBM é, um *standard* para eventos a serem utilizados pela gestão da empresa e por aplicações de negócio. O formato CBE define a estrutura do evento num formato consistente comum, sendo esse formato expresso como um documento XML, facilitando a intercomunicação entre componentes empresariais díspares. Este formato é extensível e, além das propriedades do evento

padrão, um evento pode também conter elementos de dados estendidos (Ogle, Kreger, & Salahshour, 2004).

- *Security Device Event Exchange*

O *Security Device Event Exchange* (SDEE) é uma especificação para os formatos de mensagem e os protocolos de mensagem, utilizados na comunicação dos eventos gerados pelos dispositivos de segurança. O SDEE foi desenvolvido pelo grupo de trabalho da ICSA Labs, denominado de *Intrusion Detection Systems Consortium* (IDSC) que consistia em entidades como a Cisco, Fortinet, INFOSEC Technologies, ISS, SecureWorks, SourceFire, Symantec e Tripwire. O SDEE foi projetado para ser flexível e extensível de modo a que os fornecedores possam utilizar extensões específicas de produtos de forma a manter a compatibilidade das mensagens. Baseia-se nos *standards* XML, *Hypertext Transfer Protocol* (HTTP), e no *Secure Sockets Layer*/*Transport Layer Security* (SSL/TLS) de modo a facilitar a adoção pelos vendedores e utilizadores, permitindo-lhes utilizar *software* existente que implemente essas interfaces *standard* (Fry & Nystrom, 2009).

- *WebTrends Enhanced Log File Format*

O *WebTrends Enhanced Log File Format* (WELF) é semelhante ao CEF no sentido em que não está vinculado a qualquer revendedor específico para o transporte e representação dos formatos. WELF Consiste em vinte e quatro campos, obrigatórios e opcionais, e é uma sintaxe limitada para expressar eventos de *firewalls*, redes virtuais privadas (VPN), e outros eventos baseados em rede simples (Heinbockel, Judge, McQuaid, Chuvakin, & Marty, 2008).

- *Distributed Audit Service (XDAS)*

O XDAS foi um projeto *standard* publicado em 1997 pelo Open Group. A especificação é bastante grande e tem a intenção de resolver o problema do *Log Exchange* através da definição de *Application Programming Interfaces (APIs).* A principal visão do novo *standard* XDAS é ser algo que defina um formato interoperável de relatórios de eventos, de modo a que possam ser utilizados de igual forma por todos que necessitem ter conhecimento.

- *Log Event Extended Format*

O *Log Event Extended Format* (LEEF) é um formato personalizado para a tecnologia IBM Security QRadar, que contém de forma simplificada e legível, os eventos processados para o QRadar. O formato LEEF consiste num cabeçalho opcional *syslog*, um cabeçalho LEEF e uma coleção de atributos que descrevem o evento.

- *CEE-Enhanced Syslog*

CEE-Enhanced Syslog é o próximo padrão para expressar dados estruturados dentro das mensagens *logs* (Gerhards, 2012).


```
@cee: {"source": "machine.local", "nteventlogtype": "Security",
"sourceproc": "Microsoft-Windows-Security-Auditing", "id":
"4648", "categoryid": "12544", "category": "12544",
"keywordid": "0x8020000000000000", "user": "N\\A",
"SubjectUserSid": "S-1-5-11-222222222-333333333-4444444444-
5555", "SubjectUserName": "User", "SubjectDomainName":
"DOMAIN", "SubjectLogonId": "0x5efdd", "LogonGuid": "{00000000-
0000-0000-0000-000000000000}", "TargetUserName":
"Administrator", "TargetDomainName": " DOMAIN ",
"TargetLogonGuid": "{00000000-0000-0000-0000-000000000000}",
"TargetServerName": "servername", "TargetInfo": " servername ",
"ProcessId": "0x76c", "ProcessName":
"C:\\Windows\\System32\\spoolsv.exe", "IpAddress": "-",
"IpPort": "-", "catname": "Logon", "keyword": "Audit Success",
"level": "Information"}
```


Figura 8 - Exemplo do formato *CEE-Enhanced Syslog* (Fonte: Gerhards, (2012))

Na Figura 8 é possível visualizar o exemplo de uma mensagem do formato proposto. Basicamente, o *CEE-Enhanced Syslog* é um formato de mensagem opcional e estendido para o formato *syslog*, que em conjunto com o CEE serve de apoio, por exemplo, para os *logs* do *Windows.* Dentro da mensagem *syslog* um *cookie* especial (“@cee:”) é seguido por uma representação de dados em *JSON.*

- *Common Log Format, Extended Common Log Format e Extended Log Format*

Os servidores web geram *logs* num formato *standard*, nomeadamente os criados pela entidade W3C e pela *National Center for Supercomputing Applications* (NCSA). O *Common Log Format* (CLF) e o *Extended Common Log Format* (ECLF), também designado por *Combined Log Format*, foram criados pela NCSA e *o Extended Log Format* (ELF) pela entidade W3C. O CLF é um tipo de log de formato fixo ASCII, como tal não pode ser alterado ou adaptado. Está disponível para *Web* e para serviços SMTP e NNTP. Foi criado com a intenção de acabar com os formatos proprietários tornando-se assim no primeiro *standard* na área. O ECLF é o resultado do CLF por acréscimo de novos campos. O ELF foi criado com a intenção de obter um *standard* que satisfizesse as necessidades dos clientes e é passível de ser parametrizado (Guimarães & Marques, 1992).

3.5 Trabalhos Relacionados

Hammoud afirma que o maior dos problemas enfrentados pelos administradores de sistemas é a integração e correlação de *logs* distribuídos e heterogéneos, pois todos esses *logs* podem identificar uma ameaça, e os administradores por dia deparam-se com centenas ou milhares de registos. No seu contributo para a correlação de eventos identifica as seguintes propriedades dos *logs*:

- Os ficheiros de *logs* são geralmente distribuídos e enormes;
- Eventos *logs* são mal estruturados e contém um texto robusto;
- Os ficheiros de *logs* são geralmente heterogéneos. Se um evento for identificado, não existe uma forma simples de verificar possíveis relações com outros eventos que dizem respeito a outros componentes do sistema;
- Entradas dos arquivos de *logs* são um fluxo contínuo;
- Os ficheiros de *logs* contêm eventos com diferentes padrões.

Apesar de não especificar como o faz, inclui um mecanismo responsável por transformar o ficheiro de *logs* num formato comum. O *wrapper* é a entidade responsável pela transformação dos *logs* de entrada num formato comum e está parametrizado para entradas como o *syslog*, *log4j* e outros (Hammoud, 2009).

Na análise forense o problema dos formatos de *logs* também está evidenciado. Existe um enorme volume de formatos uniformes e com redundância, incomportável para análise forense (Lin, Zhitang, & Cuixia, 2009). Os mesmos autores no seu *framework* apresentado na Figura 9 utilizam uma componente denominada "f*ormalization"* em que armazenam os eventos numa base de dados de eventos de baixo nível, após os procedimentos de normalização e padronização dos formatos.

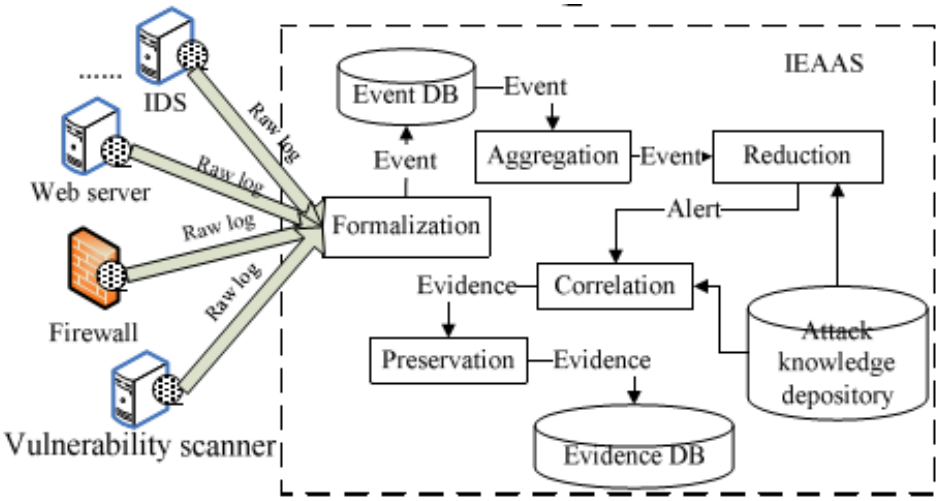


Figura 9 - *Framework* IEAAS (Fonte: Lin et al. (2009))

Madani et al, na sua sugestão de arquitetura ilustrada na Figura 10, menciona alguns desafios significativos no âmbito da gestão de *logs*, nos quais o problema da transferência dos formatos *logs* está inserido.

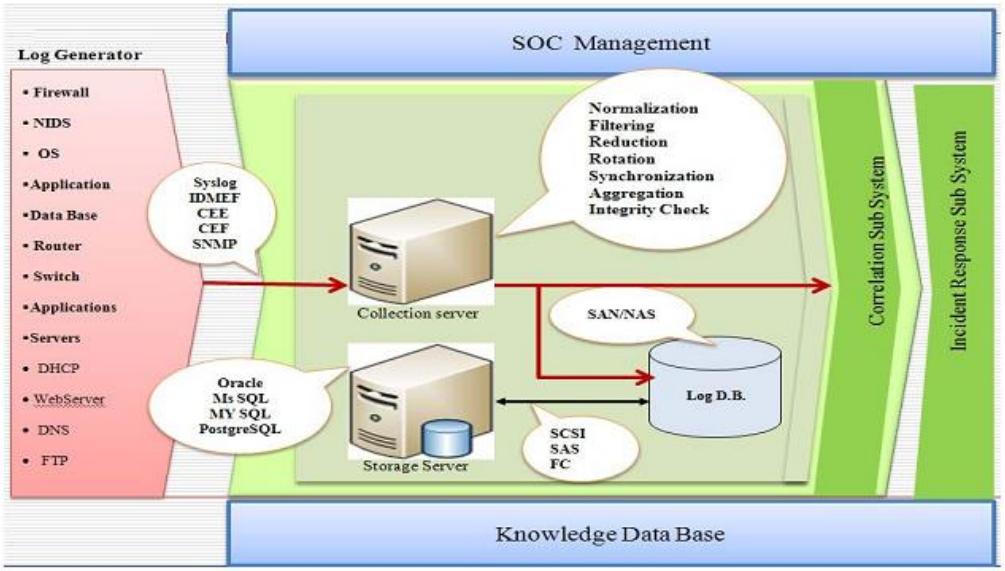


Figura 10 - Arquitetura gestão de *logs* (Fonte: Madani, Rezayi, & Gharaee (2011))

O servidor de recolha é o primeiro módulo da arquitetura a ter contato com os *logs* provenientes da *firewall*, do sistema operativo, entre outros. Os produtores de *logs* enviam *logs* utilizando protocolos como IDEMF, CEE, CEF e SNMP. O servidor de recolha deve ser capaz de compreender todos os formatos. Contudo a produção e armazenamento de *logs* é uma tarefa complicada devido a vários fatores como o número elevado de registos, a inconsistências dos conteúdos dos *logs*, formatos, e *timestamp* (Madani, Rezayi, & Gharaee, 2011).

Nos sistemas IDS, os *logs* também têm sido alvos de especial atenção e o problema subsiste no que diz respeito à diversidade dos formatos. Na arquitetura apresentada na Figura 11, o autor tem o cuidado de em locais diferentes ter unidades de processamento responsáveis pela captura dos eventos registados, que é o passo inicial para todo o processo. Como é percetível na arquitetura, cada unidade de processamento contém um conversor XML, um mecanismo de extração de recursos, um mecanismo de comparação e uma base de conhecimento do comportamento normal.

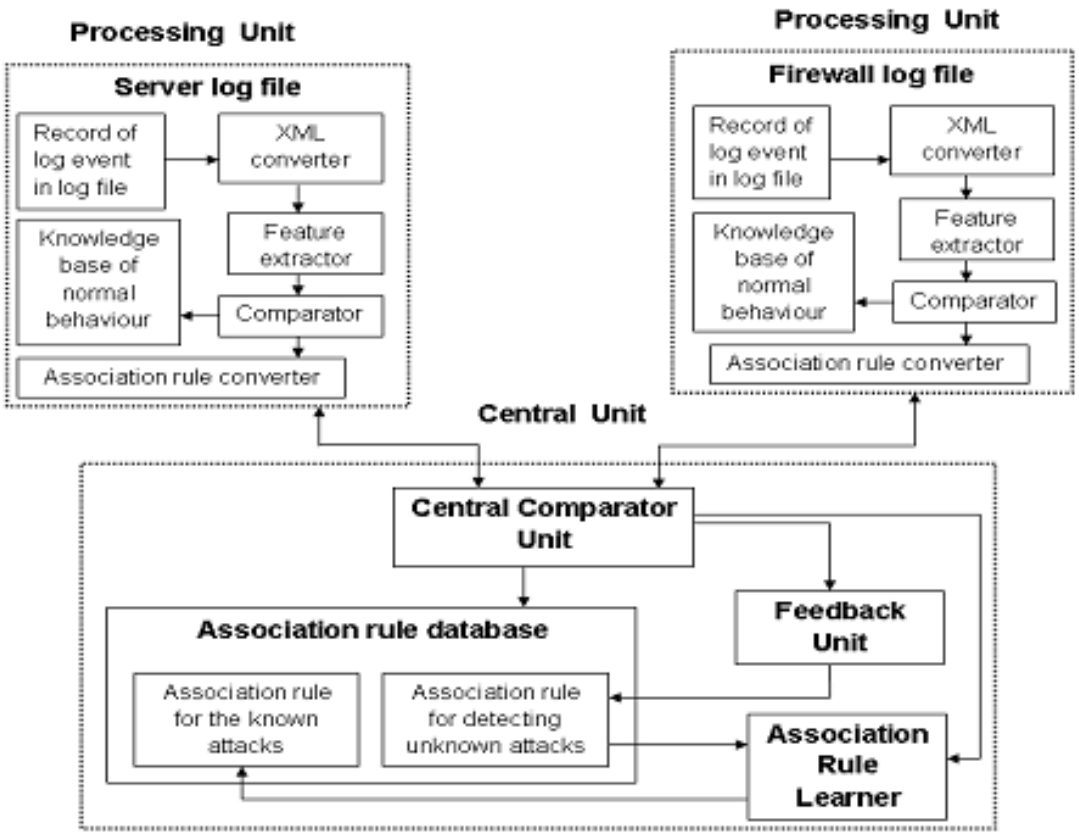


Figura 11 - IDS utilizando ficheiros *log* (Fonte: Deokar & Hazarnis (2012))

A unidade de processamento, alvo de especial atenção, é o XML *converter*. O *XML converter*, converte cada novo evento traçado de um ficheiro de *log* para o formato XML (Deokar & Hazarnis, 2012). O autor incide a sua escolha no formato XML em detrimentos dos arquivos de texto, pois segundo o próprio oferece melhor desempenho. Outros autores mencionam que o XML segue um formato estruturado e é um formato mais legível pelas máquinas (Salama et al., 2011).

3.6 Necessidade de um formato universal

Os *logs* são a chave de compreensão para muitos dos problemas que acontecem nas organizações, relacionados com a segurança de informação. Os *logs* estabelecem-se então, como uma vertente importante para a segurança de informação, sendo imprescindível para todas as tecnologias que necessitem dos *logs* para a sua função de gestão e análise, nomeadamente para os SIEM. Os *logs* são imprescindíveis para os SIEM pois à medida que aumenta o número de *logs* disponível, aumenta o potencial do SIEM para enriquecer a informação útil a extrair.

O dilema em torno dos *logs*, entre outros, consiste na inexistência de um formato universalmente aceite, no facto do mesmo evento ser descrito em distintos registos *log* e em variados formatos, e para os humanos é de muita difícil interpretação. Ao longo do tempo existiram vários esforços para procurar resolver este problema, contudo, com limitado sucesso. Seria muito benéfico se cada sistema operativo, cada aplicação, cada sistema apresenta-se os *logs* no mesmo formato, mas isso infelizmente não acontece.

A resolução deste dilema passa pela normalização dos formatos de *log*. Essa normalização tenta responder a necessidade da sua uniformização, tendo como objetivo final a transformação dos *logs* em um formato padronizado que seja mais fácil de entender, quer no âmbito da interpretação humana ou computacional. A normalização dos *logs* é que vai permitir, por exemplo, a um SIEM pesquisar mais eficientemente informação em vários dispositivos e correlacionar eventos entre eles. Torna também mais fácil a correlação automática desses eventos, por exemplo, a correspondência dos campos entre os eventos dos *logs* em intervalos de tempo.

4. Análise dos Eventos

De modo a analisar em maior detalhe o formato dos eventos de *logs*, neste capítulo realiza-se uma contextualização dos formatos de representação existentes em diversos sensores, e em particular sobre os atributos que os constituem. Esta abordagem pretende aumentar o conhecimento em torno dos eventos de segurança de informação, de modo a auxiliar no processo de tomada de decisão na criação do formato de representação de eventos de segurança de informação, bem como na proposta da *interface* que sustentará esse formato. Esta análise é composta tanto por abordagens teóricas como por abordagens práticas.

4.1 Análise dos formatos

A estratégia utilizada para a análise dos formatos de representação existentes centra-se no estudo em torno das tecnologias que compõe o OSSIM e o *Security-Onion*.

A ferramenta *Security-Onion* é uma distribuição Linux, baseada em *Ubuntu* direcionada para os IDS, *Network Security Monitoring* e a gestão de *logs*. Este conjunto de componentes centrais entrelaça perfeitamente em três funções básicas: 1) captura de pacotes (*netsniff-ng*), 2) função NIDS (*Snort*, *Suricata*, *Bro-IDS*, entre outros), HIDS (OSSEC) e 3) poderosas ferramentas de análise (Sguil, Squert, Snorby, entre outros).

Esta análise foi realizada através da documentação existente das ferramentas que constituem o OSSIM e o *Security-Onion*, com o objetivo de identificar como se comportam as tecnologias em relação aos formatos existentes. Para auxiliar está análise é criada uma tabela com os diversos sensores e identificado o formato de representação que o constitui cada um desses sensores. Após a análise, é possível concluir o seguinte:

- Existe uma diversidade enorme de formatos;
- Os mesmos sensores têm formatos diferentes;
- Os formatos variam mediante o tipo de sensor (*ids*, *netflows*, servidores *web*, sistemas operativos, *firewall*, entre outros);
- O formato utilizado em um maior número de vezes, além do formato proprietário, é o *syslog*. Apesar de muitas das vezes não ser o formato de representação pré-definido, a maior parte dos sensores permite transformações para este formato.

Esta análise vai de encontro às conclusões retiradas aquando a revisão da literatura, que identifica a enorme diversidade de formatos existentes como uma entrave à gestão dos *logs.*

4.2 Análise dos atributos

Com vista a obter um nível conhecimento mais aprofundado sobre os atributos existentes, foram realizadas 4 tipos de análises: 1) Análise dos atributos dos 14 sensores, 2) Análise dos atributos através da documentação do OSSIM, 3) Análise técnica dos atributos do OSSIM, 4) Análise dos atributos do OSSEC. Estas 4 análises são apresentadas de seguida:

- **Análise dos atributos dos 14 sensores**

Esta abordagem inicial visa conhecer que atributos estão inerentes a um conjunto de sensores mais comummente utilizados na segurança da informação. O cuidado tido nesta análise, apesar do reduzido número de sensores considerados, foi o de abranger várias áreas da segurança (com um maior grau de incidência na área da rede), de modo a obter uma perspetiva geral dos atributos. Para o efeito foram selecionados sensores do tipo: *Honeypot, Service Server, Network Monitoring, Netflows, Network Mapper, Firewall, Routers & VPN, IDS e Network Activity Audity*.

A partir destes 14 sensores foi possível identificar 103 atributos diferentes. Destes atributos, os considerados mais relevantes (neste caso concreto, no sentido de serem utilizados mais vezes) são os 20 apresentados na Tabela 3.

Tabela 3 - Resultado parciais da análise 1- Atributos relevantes dos formatos de representação de eventos

Attributes	Description	Total
timestamp	Momento em que o evento ocorreu	14
src_ip	Endereço de IP de origem	13
src_port	Porta de origem	10
dst_port	Porta de destino	10
dst_ip	Endereço IP de destino	10
protocol	Tipo de protocolo	10
interface	Interface de rede	6
size_packet	Tamanho do pacote	6
tcp_flags	Informação sobre as bandeiras TCP	5
type_service	Descrição do tipo de serviço	5
mac_Src	Endereço Mac de origem	5
transmited_packets	Número de pacotes transmitidos	4
delta_timestamps	Diferença entre timestamp	4
comment	Informação adicional	4
mac_dst	Endereço Mac de destino	4
hostname	Identificador único na rede	4
timestamp_end	Tempo em que o evento acabou	3
icmp_code	Código ICMP	3
alert_type	Tipo de alerta	3
status	O estado do evento	3

A coluna designada “Total” diz respeito ao número de sensores cuja saída contém o respetivo atributo. Os restantes atributos identificados (ou seja, 83) são comuns num número menor ou igual a duas vezes.

- **Análise dos atributos através da documentação do OSSIM**

O OSSIM define quatro tipos de eventos que são tratados de maneira diferente, dependendo do seu tipo de dados. Os tipos de eventos são os seguintes: *Normalized Event*, *Mac Event*, *OS Event* e *Service Event*. Cada um destes tipos é descrito de seguida.

- *Normalized Event*

Os atributos do *Normalized Event* (ver Tabela 4) são gerados por diferentes *plugins* ou dispositivos. Os *plugins* são cada um dos elementos definidos pelo agente (componentes responsáveis por recolher toda a informação enviada pelos diversos dispositivos existentes na rede), de forma a analisar e a normalizar as informações provenientes de um determinado dispositivo.

Tabela 4 - Atributos do *Normalized Event* do OSSIM (Adaptado: Miller & Pearson, (2011))

Attributes	Description
type	Tipo de evento: Detetor ou monitor
date	Data em que o evento foi gerado
sensor	Endereço IP do sensor que gerou o evento
interface	Nome da interface de rede associada com o evento
plugin_id	Identificador da origem de dados que gerou o evento
plugin_sid	Atribuído pelo servidor ossim
priority	Prioridade do evento (utilizado no cálculo de risco)
protocol	Protocolo de comunicação utilizado (TCP, UDP, ICMP, etc.)
src_ip	Endereço de IP de origem do evento gerado
src_port	Porta de origem do evento gerado
dst_ip	Endereço de IP de destino do evento gerado
dst_port	Porta de destino do evento gerado
log	A entrada do *log* original
data	Pode ser utilizado para guardar dados específicos do *plugin*
username	Utilizador que gerou o evento, utilizado principalmente em HIDS
password	Palavra passe utilizada no evento
filename	Ficheiro utilizado no evento, utilizado principalmente em HIDS
userdata1 to userdata9	Estes oito campos podem ser utilizados para qualquer tipo de dados

- *Mac Event*

Representam os atributos dos eventos que informam sobre mudanças no endereço MAC para endereços de IP específicos. Esses atributos estão descritos na Tabela 5.

Tabela 5 - Atributos do OSSIM do tipo *Mac Event* (Adaptado: Miller & Pearson, (2011))

Attributes	Description
Host	*Host* IP no qual foi alterado o Mac
Date	Data em que o evento foi gerado
sensor	Endereço IP do sensor que gerou o evento
interface	Nome da interface de rede associada com o evento
plugin_id	Identificador da origem de dados que gerou o evento
plugin_sid	Atribuído pelo servidor ossim
mac	Endereço Mac em hexadecimal
vendor	Fabricante da placa
log	Dados do evento que o *plugin* específico considera como parte do registo, e que não está presente em outros campos
userdata1	Cópia do endereço Mac
userdata2	Cópia do fabricante da placa
userdata3 to userdata9	Estes campos podem ser definidos pelo utilizador a partir do *plugin*. Contém qualquer tipo de informação alfanumérica

- *OS Event*

Neste tipo de evento é possível obter informação sobre alterações no sistema operativo de uma máquina. Os atributos definidos no OSSIM relativamente a este tipo de evento são descritos na Tabela 6.

Tabela 6 - Atributos do OSSIM do tipo *OS Event* (Adaptado: (Miller & Pearson, 2011))

Attributes	Description
host	IP para o sistema operativo que foi apresentado
date	Data em que o evento foi gerado
sensor	Endereço IP do sensor que gerou o evento
interface	Obsoleto
plugin_id	Neste caso será sempre o id 1511 (p0f)
plugin_sid	Atribuído pelo servidor ossim
os	Sistema operativo apresentado para o host indicado
log	Dados do evento que o *plugin* específico considera como parte do registo, e que não está presente em outros campos
userdata1	Neste caso é utilizado para a correlação
userdata2 to userdata9	Estes campos podem ser definidos pelo utilizador a partir do *plugin*. Contém qualquer tipo de informação alfanumérica

- *Service Event*

A utilização destes eventos servem para obter um inventário dos sistemas existentes na rede, bem como por exemplo aplicações novas ativas e portas abertas. Os atributos são descritos na Tabela 7.

Tabela 7 - Atributos do OSSIM do tipo *Service Event* (Adaptado: (Miller & Pearson, 2011))

Attributes	Description
host	*Host* IP
date	Data em que o evento foi gerado
sensor	Endereço IP do sensor que gerou o evento
interface	Obsoleto
plugin_id	Este será geralmente 1516 (pads)
plugin_sid	Atribuído pelo servidor ossim
port	Apresenta a porta aberta na máquina host
log	Dados do evento que o plugin especifico considera como parte do registo, e que não está presente em outros campos
protocol	Protocolo de comunicação utilizado (TCP, UDP, ICMP, etc.)
service	Tipo de serviço existente na porta especificada
application	Aplicação que executa o serviço exibido
userdata1	Cópia do campo aplicação
userdata2	Cópia do campo serviço
userdata3 to userdata9	Estes campos podem ser definidos pelo utilizador a partir do *plugin*. Contém qualquer tipo de informação alfanumérica

- **Análise técnica dos atributos do OSSIM**

O estudo destes atributos tem como base os dois tipos de *plugins* definidos pelo OSSIM (Detetor e Monitor) e foi realizado através de uma análise detalhada dos ficheiros que contêm os parâmetros de configuração dos *plugins* e das regras a que cada evento tem de corresponder para ser recolhido e normalizado. Sucintamente, os *plugin* do tipo Detetor são responsáveis por fornecer os eventos associados aos componentes *Snort*, *firewall*, antivírus, eventos do sistema operativo, entre outros, e os do tipo Monitor são responsáveis por fornecer informações do evento sob a forma de indicadores (*Ntop*, *tcptrack*, *nmap*, entre outros).

Nesta análise além da identificação dos atributos, é também registado o tipo de sistema em que esse sensor se insere (*Honeypot*, *IDS*, *Firewall*, dispositivos de rede, entre outros), o tipo de evento (detetor ou monitor), a origem (*log*, *Mssql Database*, *MySql Database*, *Windows Management Instrumentation*), as regras associadas e a localização onde as aplicações registam os eventos (por exemplo /var/log/syslog).

No final, é possível obter um mapa dos atributos inerentes a cada sensor, de onde é possível deduzir quais os atributos mais frequentes nos sensores do mesmo tipo, quais os atributos comuns entre os sensores de diferentes tipos, quais os atributos que são sempre utilizados e obrigatórios, bem como os que são utilizados opcionalmente. Na Figura 12 é apresentada uma imagem parcial da tabela construída.

Attributes	Optional	Traffic Devices (19)	Honeypot (7)	Service Server (21)	Virus (6)	Mail (5)	Web Proxy (2)	IDS (7)	Firewall (18)	VPN (4)	IPS (6)	Total
Plugin_SID	0	19	7	21	6	5	2	7	18	4	6	156
Normalize Date	1	19	5	21	3	5	2	6	17	3	6	135
Source IP	0	13	7	19	2	5	2	7	18	4	5	124
Destination IP	0	16	6	11	0	3	2	5	17	4	5	95
Sensor	0	15	1	13	1	4	1	3	12	2	3	79
Source Port	0	7	6	8	0	1	0	4	18	1	6	64
Destination Port	1	6	6	8	0	2	2	5	15	3	4	62
Username	0	9	1	14	1	1	1	0	4	4	2	61
Protocol	3	3	2	2	0	0	1	5	15	0	2	37
Message	15	3	0	4	0	0	0	2	5	0	1	37
Filename	0	1	0	7	4	0	0	0	0	1	0	23
Interface	6	3	0	1	0	0	1	3	5	0	1	19
Service Name	11	2	1	3	0	0	1	0	1	0	1	13
Type	9	2	2	0	0	0	0	0	2	1	0	12
Severity	7	4	0	1	1	0	0	1	0	0	0	12

Figura 12 – Vista Parcial da Análise 3 – Mapeamento dos atributos presentes nos sensores do OSSIM

Relativamente à Figura 12 na primeira coluna estão registados os atributos dos sensores, no cabeçalho da tabela os tipos de sistema, por grupos, e a última coluna diz

respeito ao número total de vezes que esse atributo é utilizado. No cabeçalho da tabela está ainda referenciado, entre parênteses, o número de sensores associados por cada tipo de sistema (por exemplo, 19 sensores do tipo *Traffic Devices*, 7 sensores do tipo *Honeypot*, 18 sensores do tipo *Firewall*).

De modo a ilustrar a forma como a tabela constante na Figura 12 foi obtida, é apresentada de seguida a análise de três atributos:

- *Plugin_Sid*: Este atributo está presente em todos os sensores (156/156).
- *Protocol*: Este atributo está presente em 37 sensores num total de 156. Pela análise da tabela parcial, é possível verificar que este atributo não está presente em todos os tipos de sistema (por exemplo V*PN*, *Virus* e *Mail* tem o valor igual a 0) nem está presente em todos os sensores do mesmo tipo de sistema (por exemplo em 19 sensores do tipo *Traffic Devices* este atributo só está presente 3 vezes).
- *Sensor*: Este atributo está presente 79 vezes num total de 156 sensores. Pela tabela parcial é possível verificar que está presente em todos os grupos por tipo de sistema disponíveis, apesar de não estar presente em todos os sensores do mesmo tipo (por exemplo apenas está presente em 13 sensores dos 21 do tipo de sistema *Service Server*).

No total foram analisados 156 ficheiros de configuração, foram identificados 44 tipos e 292 atributos.

- **Análise dos atributos do OSSEC**

O OSSEC é uma ferramenta HIDS *open source*, multiplataforma, orientada para análise e correlação de *logs*, monitorização e análise de *firewall*, *ids*, servidores *web*, registo do *Windows* e autenticação de *logs* (Bray, Cid, & Hay, 2008a). Os atributos são descritos na Tabela 8.

Tabela 8 - Resultado dos atributos do OSSEC (Adaptado: (Bray, Cid, & Hay, 2008b))

Attributes	Description
id	Identificador do evento
hostname	Hostname da origem do evento
srcip	Endereço de IP de origem do evento gerado
dstip	Endereço de IP de destino do evento gerado
srcport	Porta de origem do evento gerado
dstport	Porta de destino do evento gerado
protocol	Protocolo de comunicação utilizado (TCP, UDP, ICMP, etc.)
action	Ação realizada no evento
srcuser	Utilizador de origem inserido no evento
dstuser	Utilizador de destino inserido no evento
status	O estado do evento
command	O comando que foi chamado dentro do evento
url	O url associado ao evento
data	Qualquer tipo de dados adicionais
systemname	O nome do serviço associado ao evento
program_name	Nome da aplicação. É retirada do cabeçalho do syslog do evento
log	Secção das mensagens do evento
full_log	O evento completo
location	Local de onde o log veio

Em síntese, a análise destas quatro abordagens permite demonstrar que: a) existe um elevado número de atributos utilizados para caracterizar eventos de segurança; b) diferentes atributos podem ter múltiplas interpretações; c) o mesmo atributo pode ter diferentes terminologias; e d) diferentes tecnologias utilizam diferentes atributos para caracterizar um evento de segurança.

Esta página foi colocada propositadamente em branco.

5. Especificação e Implementação do Formato de Representação de Eventos e Interface

Neste capítulo é apresentado o contributo em torno dos eventos de segurança da informação. Em particular, pretende-se formalizar a proposta para o formato de representação de eventos de segurança da informação e também uma proposta para uma *interface* de representação de eventos.

5.1 Abordagens de decisão

No desenvolvimento da proposta do formato, são utilizadas 3 abordagens, nomeadamente: 1) agrupamento por classes, 2) frequência dos atributos e 3) grupos ou domínio do evento origem. Estas abordagens têm como base a informação recolhida nas 4 análises apresentadas anteriormente no capítulo 4.

- Agrupamento por Classes

As classes foram definidas aquando do levantamento dos atributos presentes nos ficheiros de configuração dos sensores que constituem o OSSIM e incide na função dos sensores. Cada classe é definida mediante o tipo de sensor e corresponde a um ou mais sensores.

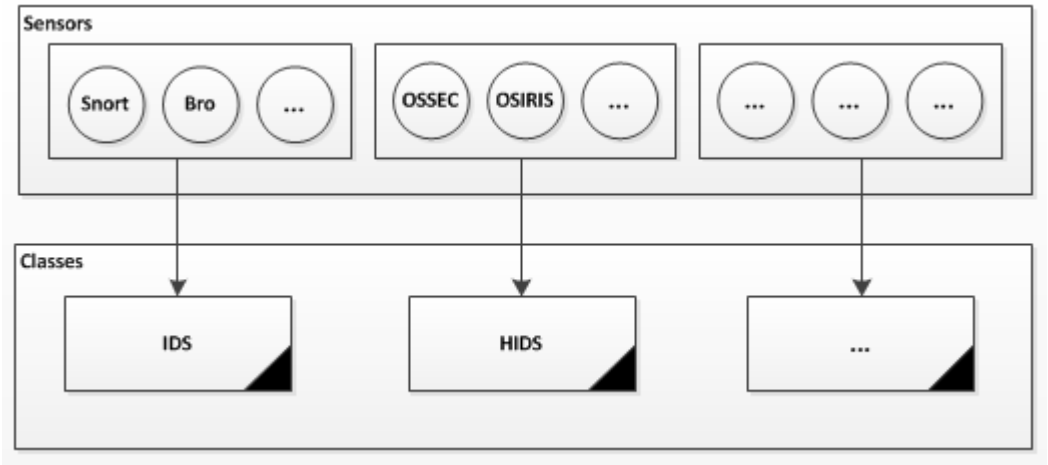


Figura 13 - Exemplo de agrupamento por classes

Como se verifica na Figura 13, existe um conjunto de sensores associados a uma determinada classe. Neste exemplo e por questões de facilidade de visualização, apenas são apresentadas duas classes, no entanto estão definidas no total 44 classes (ver Anexo 1.). O resultado desta abordagem permite ter uma visão alargada sobre os atributos inerentes a cada classe de eventos, bem como os atributos comummente utilizados entre diferentes classes, sendo possível identificar diversos padrões. Como se pode constatar na Figura 14, é utilizado um esquema de cores que serve como auxílio na interpretação do mapa visual. Sempre que possível, são utilizadas cores distintas para cada atributo, à exceção da cor encarnada, que tem como finalidade sinalizar essa propriedade como "não presente" em determinada classe.

	Attributes									
Source	Source IP	Destination IP	Sensor	Source Port	Destination Port	Username	Protocol	Message	Filename	Interface
Admin System/Security	1	1	1	0	0	1	0	1	0	0
Agentless Monitoring	0	1	1	0	0	0	0	0	0	0
AntiMalware	1	0	0	0	0	0	0	1	0	0
AntiSpam	1	1	1	1	1	1	0	1	0	0
Antivirus	1	0	1	0	0	1	0	0	1	0
Authentication	1	1	1	0	1	1	0	1	1	0
Camera Video	1	0	0	0	0	0	0	1	1	0
CC transaction	0	0	0	0	0	1	0	1	0	0
Central Policy	1	1	0	0	0	1	0	1	1	0
Cluster Service	1	1	1	1	0	1	0	1	1	1
HIDS	1	1	1	1	1	1	1	1	1	0
Honeypot	1	1	1	1	1	1	1	0	0	0
Host Malware	0	0	0	0	0	0	0	0	0	0
Management Software	1	0	1	0	0	1	0	1	1	0
Messaging Security	0	0	0	0	0	0	0	0	0	0
Mobile	0	0	1	0	0	0	0	0	1	0
Monitor Arp	0	0	0	0	0	0	0	0	0	1
Monitoring Network	1	1	1	1	1	0	1	1	0	1
Monitoring windows	0	0	0	0	0	1	0	0	0	0
Packet Filter	1	1	1	1	1	0	1	0	0	1
PADS	1	1	0	0	0	0	1	0	0	0
Process	1	0	0	0	0	1	0	1	1	0
Usb Control	0	0	1	0	0	0	0	0	0	0
Virtualization	1	0	1	1	0	0	0	1	1	0
VPN	1	1	1	1	1	1	1	0	1	0
Vulnerability Scanner	1	1	1	0	1	1	1	1	0	0
WAF	1	1	0	0	1	0	1	0	0	0
Web Aplication	1	0	0	0	0	1	0	1	0	0
Web Proxy	1	1	1	0	1	1	1	0	0	1
Web Security	1	1	1	1	1	1	0	1	0	0
Ws eventlog-to-syslog	1	0	0	0	0	0	0	0	0	0

Figura 14 – Vista parcial da tabela (Parte I)

À medida que o nível de detalhe da classe vai aumentando em termos de especificidade, os atributos comuns entre as diferentes classes vão diminuindo, como é passível ver na Figura 15.

	Attributes												
Class	Md5	Status	Group	Domain	Process	ID	Referer	Policy	Size	UserAgent	User	Event	Virus
Admin System/Security	0	0	0	0	0	1	1	0	0	0	0	0	0
Agentless Monitoring	0	1	0	0	0	0	0	0	0	0	0	0	0
AntiMalware	0	0	0	0	0	0	0	0	0	0	0	0	0
AntiSpam	0	0	0	1	0	0	0	0	0	0	1	0	0
Antivirus	0	0	0	0	0	0	0	0	0	0	0	0	1
Authentication	1	1	1	0	1	0	0	0	0	0	1	0	0
Camera Video	0	0	0	0	0	0	0	0	0	0	0	0	0
CC transaction	0	0	0	0	0	0	0	0	0	0	0	0	0
Central Policy	0	0	0	0	0	0	0	0	0	0	0	0	0
Cluster Service	0	0	0	0	1	0	0	0	0	0	0	0	0
HIDS	0	0	0	1	0	0	0	0	1	1	0	0	0
Honeypot	0	0	0	0	0	1	0	0	0	0	0	0	0
Host Malware	0	0	0	0	0	0	0	0	0	0	0	0	0
Management Software	0	0	0	0	0	0	0	0	0	0	0	0	0
Messaging Security	0	0	0	0	0	0	0	0	0	0	0	0	0
Mobile	0	0	0	0	0	0	0	0	0	0	0	0	0
Monitor Arp	0	0	0	0	0	0	0	0	0	0	0	0	0
Monitoring Network	0	0	0	0	0	0	0	0	0	0	0	1	0
Monitoring windows	0	0	0	0	0	0	0	0	0	0	0	0	0
Packet Filter	0	0	0	0	0	0	0	0	0	0	0	0	0
PADS	0	0	0	0	0	0	0	0	0	0	0	0	0
Process	0	0	0	0	0	0	0	0	0	0	0	0	0
Usb Control	0	0	0	0	0	0	0	0	0	0	0	0	0
Virtualization	0	0	0	0	1	0	0	0	0	0	0	0	0
VPN	0	0	0	0	0	0	0	1	0	0	0	0	0
Vulnerability Scanner	0	0	0	0	0	0	0	0	0	0	0	0	0
WAF	1	1	0	0	0	0	1	0	0	0	0	1	0
Web Aplication	0	0	0	0	0	0	0	0	0	0	0	0	0
Web Proxy	1	1	1	0	0	1	1	0	1	1	0	0	0
Web Security	0	0	0	0	0	1	0	0	0	0	0	0	0
Ws eventlog-to-syslog	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 15 - Vista da tabela parcial (Parte 2)

Através desta quadro é possível concluir que apesar de haver um número significativo de atributos comuns, passíveis de agregação pelos padrões identificados, existe ainda um elevado número de atributos heterogéneos e com funções e /ou propósitos muito diferentes, que dificultam e tornam mesmo impossível a definição de um formato único tendo em conta apenas esta abordagem.

- Frequência dos atributos

De modo a obter uma perceção de quais os atributos considerados mais importantes, é calculado o peso de cada atributo nas classes identificadas. Para o efeito, é utilizada uma classificação em 4 níveis: Pouco Significativo (0 a 24,99 % inclusive), Significativo (25 a 49,99 % inclusive), Muito Significativo (50 a 74.99 % inclusive) e Obrigatório 75 a 100 % inclusive). Por exemplo, como é possível ver na Figura 16, se a classe IDS tem 7 sensores e o atributo "*Source IP*" está presente nos 7, este atributo tem um peso de 100%. Esta classificação é realizada em todos os atributos identificados.

Classe	Source IP	Destination IP	Sensor	Source Port	Destination Port	Username	Protocol	Message
HIDS (2)	100%	100%	100%	50%	50%	50%	50%	50%
Honeypot	100%	86%	14%	86%	86%	14%	28%	0
Host Malware	0	0	0	0	0	0	0	0
IDM	100%	0	0	0	0	100%	0	100%
IDS	100%	71%	43%	57%	71%	0	71%	29%
IPS	83%	83%	50%	100%	67%	33%	33%	17%
iptables	100%	100%	100%	100%	100%	0	100%	100%
mail (5)	100%	60%	80%	20%	40%	20%	0	0
Management Software (1)	100%	0	100%	0	0	100%	0	100%
Messaging Security	0	0	0	0	0	0	0	0
Mobile	0	0	100%	0	0	0	100%	0
Monitor Arp (2)	0	0	0	0	0	0	0	0
Monitoring Network (3)	100%	67%	33%	33%	33%	0	33%	67%
Monitoring windows	0	0	0	0	0	100%	0	0
Packet Filter (1)	100%	100%	100%	100%	100%	0	100%	0
PADS	100%	100%	0	0	0	0	100%	0
Process	100%	0	0	0	0	100%	0	100%
Service Server (21)	90%	52%	61%	38%	38%	67%	10%	19%
Storage Management	100%	0	100%	100%	0	0	0	0

Figura 16 - Figura parcial da frequência dos atributos

Na classificação é tido em conta a disparidade do número de sensores em cada classe, bem como o número de atributos inerentes às mesmas, pois por exemplo uma frequência de 50 % numa classe com 100 atributos é diferente de uma classe com 20 atributos. Esta abordagem permite utilizar a mesma escala e classificar os atributos da mesma forma, conseguindo saber qual a sua significância na respetiva classe, e também no geral de todas as classes. Contudo, a informação proveniente desta abordagem tem de ser alvo de uma análise cuidada, pois a importância de um atributo é relativa, ou seja, por exemplo a importância pode depender da natureza do evento ou simplesmente ser um atributo muito frequente porque é fácil de encontrar e é definido por defeito. Para efeitos da presente dissertação, apenas foi alvo de análise mais cuidada a significância dos atributos na própria classe, não sendo realizado nenhum termo de comparação entre classes. Esta abordagem de forma individual pode não ser um bom método de análise, mas com a classificação adicional proposta, auxilia no processo de tomada de decisão.

- Grupos ou domínio do Evento Origem

Esta abordagem incide no agrupamento de atributos pela natureza do evento de modo a minimizar a heterogeneidade dos sistemas com propósitos e funções diferentes. São identificados 3 grupos, nomeadamente o grupo *Application Information*, *Network Information* e *System Information*. Estes 3 grandes domínios de informação resultam dos padrões identificados através das classes, da análise dos atributos pela sua frequência, da natureza do evento e através das respostas que se pretende obter, ou

seja, sobre qual o propósito do grupo, e o que é necessário saber "obrigatoriamente" para caracterizar um evento que tenha ocorrido no âmbito desse grupo.

O grupo *Application Information*, como o próprio nome sugere, diz respeito a todas as classes que têm informação relacionada com as aplicações (antivírus, aplicações web, *pads*, entre outros) o grupo *Network Information* toda a informação sobre a rede (dispositivos de rede, *netflows*, *arp*, entre outros) e o grupo *System Information* diz respeito a toda a informação relativa ao sistema (serviços, processos, utilizadores, *cluster*, sistema operativo, entre outros). Na Figura 17 é possível verificar um exemplo elucidativo de classificação por grupos de informação.

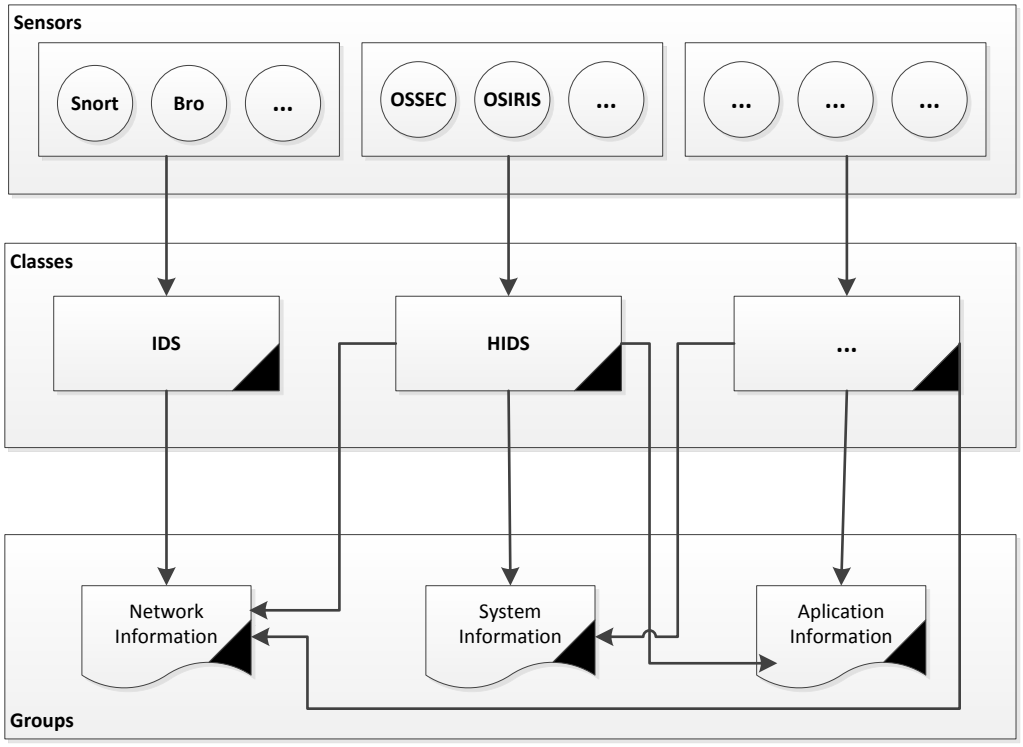


Figura 17 - Exemplo de agrupamento por grupos de informação

As classes definidas nos grupos, não são exclusivas do próprio grupo em que estão inseridas, isto é, uma classe pode estar nos 3 grupos pois pode conter atributos que acrescentem valor nos 3 domínios de informação. Por exemplo, a classe HIDS, contêm atributos considerados importantes para a rede (origem do IP, destino do IP, protocolo, entre outros), para o sistema (nome do sistema, *hostname*, processo, entre outros) e para a aplicação (nome da aplicação, versão, entre outros).

Deste modo, apesar de cada grupo ter atributos que obrigatoriamente o caracterizam, nenhum dos outros atributos pode ser descartado, pois toda a informação é importante para caracterizar um evento e, sobretudo, para poder estabelecer as correlações.

Na Tabela 9, estão apresentados os resultados do agrupamento efetuado.

Tabela 9 - Resultados da abordagem por grupos

Group	Total Class	Total Attributes
Application Information	17	45
Network Information	10	153
System Information	17	61

5.2 Estrutura do formato de representação

O resultado de um modelo *standard* de representação com base nas análises anteriores de forma individual origina um formato, que apesar de estruturado e completo, é demasiado extenso. Devido às suas características díspares resultantes da sua origem e natureza, farão com que a sua facilidade de interpretação, eficácia e eficiência de processamento sejam comprometidas.

Com base nessas premissas e de forma a contrariar as limitações expostas, a solução do formato proposto resulta da conjunção das abordagens apresentadas anteriormente. Propõe-se uma solução para um formato de representação, cuja designação é *xxUniversal LogFormat.* O *xxUniversal LogFormat* é, portanto, um formato de eventos de segurança, composto por atributos "pré-definidos" que contém eventos legíveis e fáceis de processar mediante as circunstâncias exigidas. A Figura 18 apresenta a estrutura de alto nível do formato.

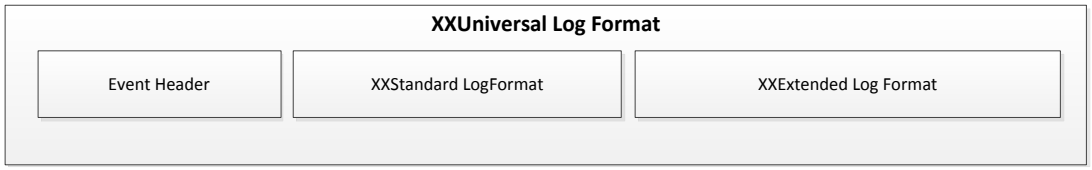


Figura 18 – Macro estrutura do formato de representação para eventos de segurança

Tal como é possível verificar, esta proposta de formato é composta por três componentes: *Event Header*, *xxStandard Log Format* e *xxExtended Log Format*.

Estas componentes têm diferentes níveis de detalhe acerca da informação do evento. Para tal, foram considerados três níveis: Nível 0, Nível 1 e Nível 2. À medida que aumenta o nível, aumenta o grau de detalhe acerca da informação do evento.

As componentes são descritas de seguida:

- *Event Header*

O *Event Header* é o identificador do evento, ou seja, contém toda a informação necessária para saber de que evento se trata (Nível 0). A finalidade do *Event Header* é de funcionar como um identificador único, sem ter a necessidade de examinar o resto da informação. Por exemplo, esta componente permite a qualquer mecanismo que necessite de informações sobre os eventos, decidir logo no início de processamento de análise, se este evento acrescenta ou não valor para o seu propósito. Os atributos definidos para o *Event Header* estão apresentados na Tabela 10.

Tabela 10- Especificação da componente *Event Header*

AttributeKey	Type	Description
EventID	String	Identificador do evento
Timestamp	Date	Tempo em que o evento foi ocorreu
GID	String	Identificador do grupo
CID	String	Identificador da classe
Sid_Event	IPv4 or Ipv6 Address	Identificador do sensor responsável por gerar o evento

Os elementos do *Event Header* são separados pelo delimitador pipe (|), de modo a distinguir esta componente das demais. Por exemplo *o Event Header* deve ser representado deste modo: |EventID|Timestamp|ClassID|SubClass|Sid_Event|.

- *xxStandard Log Format*

O *xxStandardLogFormat* tem um conjunto de atributos de eventos predefinidos que fornecem informação de alto nível sobre o evento (Nível 1).

Os atributos pré-definidos resultam da análise efetuada aos atributos comummente mais utilizados, independentemente da sua natureza ou origem, e são consideradas relevantes para o evento. Estes atributos são considerados basilares para o formato de representação, sendo considerados *standard* para dar resposta à maior parte das exigências requeridas no âmbito da gestão e análise de *logs*. Contudo, estes atributos pré-definidos não são impostos, nem muito menos limitados. Como todas as componentes, esta é extensível e permite adicionar atributos que posteriormente se considerem relevantes. Inicialmente este módulo foi idealizado para ser um módulo único que englobasse os atributos considerados mais importantes dos diversos grupos, bem como aqueles que acrescentem valor à informação do evento, como se pode consultar na Tabela 11.

Tabela 11 - Especificação dos atributos do xxStandard Log Format

AttributeKey	Type	Description
Source_IP	Ipv4 or Ipv6	Endereço de IP de origem do evento gerado
Destination_IP	Ipv4 or Ipv6	Endereço de IP de destino do evento gerado
Destination_Port	Integer	Porta de destino do evento gerado
Source_Port	Integer	Porta de origem do evento gerado
Protocol	Integer ou Keykword	Tipos de protocolo utilizados
Mac_Source	Mac Address	Endereço Mac de origem em hexadecimal
Mac_Destination	Mac Address	Endereço Mac de destino em hexadecimal
Severity	Integer	Classificação da severidade do evento
Hostname	String	Identificador de origem único do host
Url		Informação url dentro do evento
Source_User	String	Utilizador que foi identificado como criador do evento
Destination_User	String	Nome do utilizador de destino (alvo)
Sys_Info	String	Nome da aplicação, serviço, sistema, inserido no evento
Signature_Id	String	Assinatura do evento
Ext_Reference	String	Referências externas (CVE, OSVDB, Secunia, etc) que estão incluídas no evento
Domain	String	Domínio associado à conta do utilizador
Goup	String	Grupo associado à conta do utilizador
Status	String	O estado do evento (sucesso, falha, etc),
Type	String	Descrição do evento para saber do que se trata
Filename	String	Ficheiro utilizado com associação ao evento
Action	String	Ação tomada no evento (aceite, rejeitada, etc)
Command	String	O comando que foi chamado dentro do evento
Message		Secção das mensagens do evento
Data	String	Qualquer tipo de dados adicionais

Esta proposta, apesar de ser genérica e dar resposta as exigências da gestão e análise de *logs*, não será a mais adequada para uma análise lógica e eficiente, pois uma percentagem elevada dos atributos fica sem valor para muitos eventos. A decisão passa da mesma forma, pela utilização de um formato pré-definido, mas tendo em conta o grupo em que está inserido. Deste modo, dependendo do grupo, existem três *xxStandardLogFormat* diferentes, como se pode verificar na Figura 19.

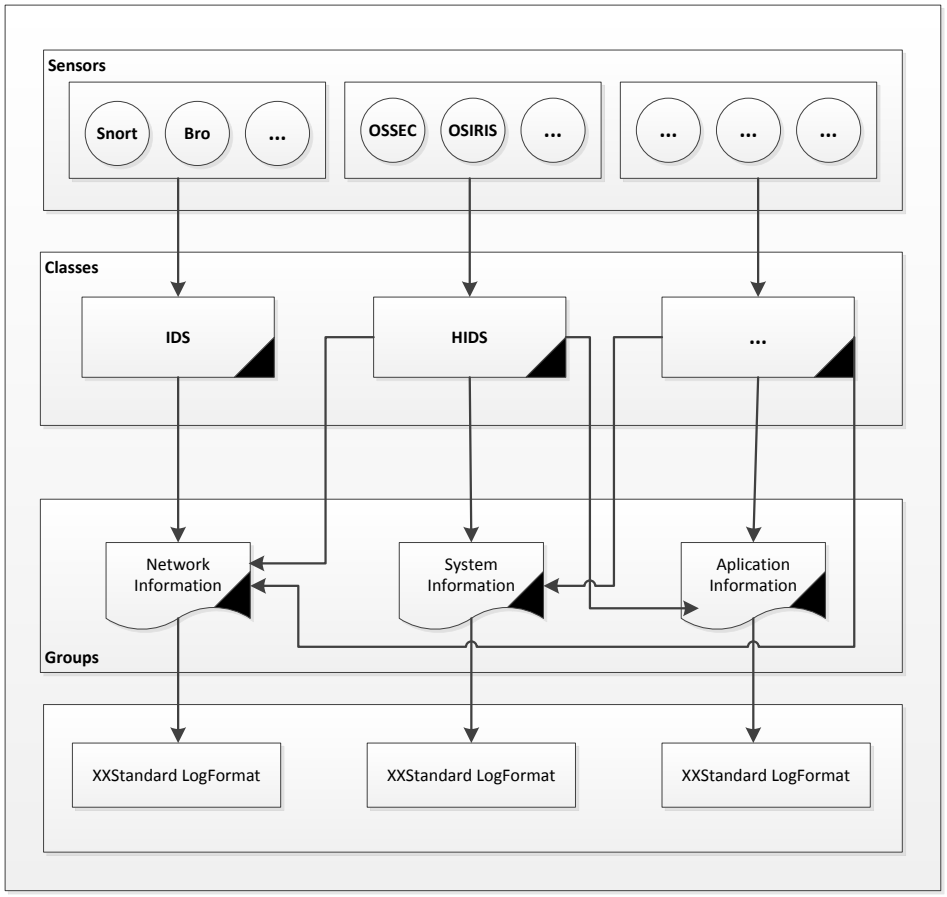


Figura 19 - xxStandard Log Format por grupo

Os atributos que caracterizam o *xxStandard Log Format*, dependendo do tipo de grupo em que estão inseridos, são apresentados nas tabelas 12, 13, 14.

Tabela 12 - *xxStandard Log Format* do grupo *System Information*

AttributeKey	Type	Description
Hostname	String	Identificador único de origem do host
Source IP	IPv4 or Ipv6	Endereço de IP de origem do evento gerado
Destination IP	IPv4 or Ipv6	Endereço de IP de destino do evento gerado
Source Port	Integer	Porta de origem do evento gerado
Source User	String	Utilizador que foi identificado como criador do evento
Destination User	String	Nome do utilizador de destino (alvo)
Service Name	String	Nome de serviço associado ao evento
Process	String	Nome do processo associado ao evento
System Name	String	Nome do sistema associado ao evento (linux, windows, mac os, etc)
Message	String	Secção das mensagens do evento
Ext_Reference	String	Referências externas (CVE, OSVDB, Secunia, etc) que estão incluídas no evento

Tabela 13 - *xxStandard Log Format* do grupo *Network Information*

AttributeKey	Type	Description
Source IP	IPv4 or Ipv6	Endereço de IP de origem do evento gerado
Destination IP	IPv4 or Ipv6	Endereço de IP de destino do evento gerado
Source Port	Integer	Porta de origem do evento gerado
Destination Port	Integer	Porta de destino do evento gerado
Protocol	Integer ou Keykowrd	Tipos de protocolo utilizados
Interface	String	Nome da interface de rede associada ao evento
Mac Source	Mac address	Endereço mac de origem em hexadecimal
Mac Destination	Mac address	Endereço mac de destino em hexadecimal
Message	String	Secção das mensagens do evento
Ext_Reference	String	Referências externas (CVE, OSVDB, Secunia, etc) que estão incluídas no evento

Tabela 14 - *xxStandard Log Format* do grupo *Application Information*

AttributeKey	Type	Description
Source IP	IPv4 or Ipv6	Endereço de IP de origem do evento gerado
Aplication	String	Nome da aplicação associada ao evento
Filename	String	Ficheiro utilizado no evento
Path	String	Localização de onde veio
Message	String	Secção das mensagens do evento
Ext_Reference	String	Referências externas (CVE, OSVDB, Secunia, etc) que estão incluídas no evento

Nesta componente, por uma questão de estrutura e boas práticas, a ordem dos atributos deve ser salvaguardada e sempre que adicionado um novo atributo, este deverá tomar o ultimo lugar da lista. De modo a distinguir esta componente, os seus atributos são separados por um delimitador "tab". Por exemplo, o *xxStandardLogFormat* deve ser representado do seguinte modo: Source_IP Destination_IP Source_Port Destination_Port.

- *xxExtended Log Format*

Esta componente contém informação de Nível 2, ou seja, é composta pelos atributos específicos do evento e acrescenta valor na medida que contém informação mais detalhada sobre o mesmo. Os atributos que compõem esta componente variam em número, dependendo do grupo e da classe do evento e é garantido que nenhuma informação sobre o evento é descurada.

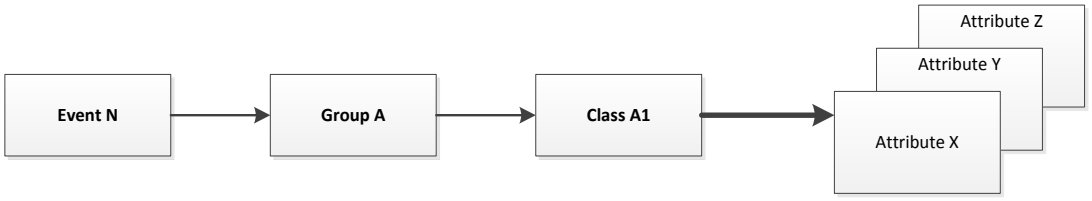


Figura 20 - Representação da estrutura do xxExtended Log Format

A informação inerente a este componente resulta de todas os restantes atributos pertencentes ao evento que não foram estabelecidos como pré-definidos para um grupo ou classe. Através da Figura 20 podemos verificar o fluxo que determina os atributos desta componente. Temos os atributos X, Y, Z do Evento N que pertence ao grupo A e que por sua vez faz parte da Classe A1. Desta forma os atributos que

constituem a componente *xxExtended Log Format* é igual aos atributos do Event N menos os atributos pré-definidos do *xxStandard Log Format* e do *Event Header*.

Por exemplo, o cenário apresentado na Figura 21 mostra um Evento N que pertence ao grupo *Network Information* e à Class ARP.

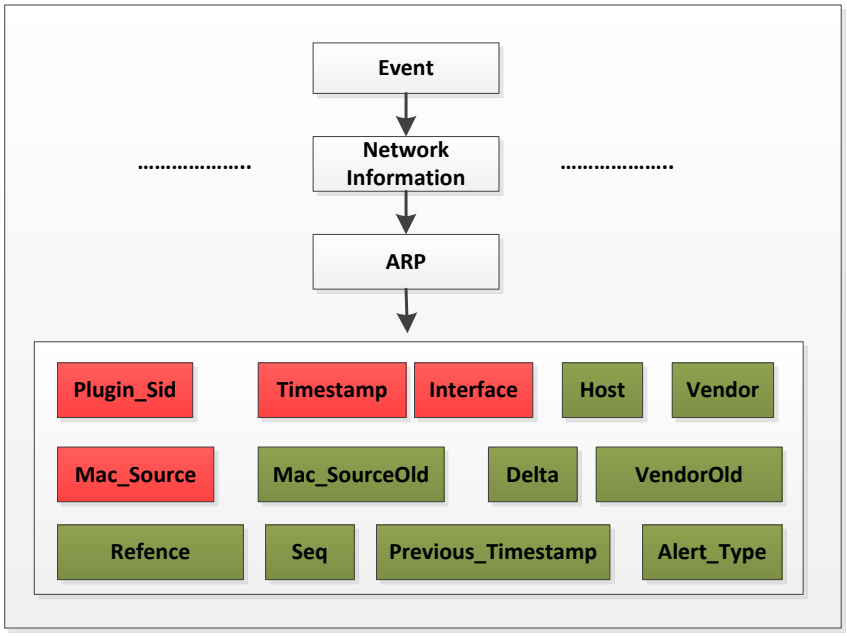


Figura 21 - Cenário dos atributos do xxExtended Log Format

Com base nesse cenário, os atributos que fazem parte da componente *xxExtended Log Format* são os atributos que estão assinalados com a cor verde. A especificação dos atributos desta componente vai depender da natureza e da origem do evento. Para este cenário, os atributos estão especificadas na Tabela 15.

Tabela 15 - Exemplo da especificação atributos do xxExtended Log Format

AttributeKey	Type	Description
Host	String	Identificador único de origem do host
Vendor	String	Nome do fabricante da placa de rede
Reference	String	Referências externas associadas ao evento
Mac_SourceOld	Mac Address	Antigo endereço mac de origem
VendorOld	IPv4 or Ipv6	Antigo nome do fabricante da placa de rede
Previous_Timestamp	Date	Antigo timestamp associado ao evento
Delta	Date	Diferença entre *timestamp*
Seq	String	Número da sequência
Alert_Type	String	Tipo de alerta associado ao evento

Cada atributo deste campo deve ser separado por um delimitador *caret* (^), sendo que a ordem dos atributos aqui não é aplicada. Por exemplo o *xxExtended Log Format* deve ser representado do seguinte modo: ^Vendor^Reference^VendorOld^.

5.3 Representação do formato log

A linguagem escolhida para a representação da estrutura do formato foi a linguagem XML, pois permite realizar uma descrição formalizada do evento de segurança, sendo consistente e compatível, e atende às necessidades de padronização de eventos de segurança em ambientes heterogéneos. Na Figura 22 é mostrada uma imagem parcial do ficheiro *XML* correspondente ao grupo de informação *Network Information*. O formato completo pode ser consultado no Anexo 2**.**

```
<?xml version="1.0" encoding="UTF-8"?>
- <Format Type="xxUniversal Log Format">
  - <Group Type="Network Information">
    - <Class Type="IDS">
      - <Module_Header>
        - <Attribute>
            <name>EventID</name>
            <Description>Identifier of the event</Description>
          </Attribute>
        - <Attribute>
            <name>Timestamp</name>
            <Description>Time at which the event occurred</Description>
          </Attribute>
        - <Attribute>
            <name>GID</name>
            <Description>Group Identification</Description>
          </Attribute>
        - <Attribute>
            <name>CID</name>
            <Description>Class Identification</Description>
          </Attribute>
        - <Attribute>
            <name>Sid_Event</name>
            <Description>Identifier of the sensor responsible for the generation of the event</Description>
          </Attribute>
        </Module_Header>
      - <Module_xxStandard_Log_Format>
        - <Attribute>
            <name>Source_IP</name>
            <Description>The source IP address within the event</Description>
          </Attribute>
        - <Attribute>
            <name>Destination_IP</name>
            <Description>The Destination_IP address within the event</Description>
          </Attribute>
```

Figura 22 - Exemplo da estrutura do formato XML (vista parcial)

Existe um ficheiro XML por cada grupo (*Network Information*, *System Information*, *Aplication Information*) que por sua vez contém as classes com os seus atributos. Dentro das classes, cada classe está dividida por um Module_Header, um Module_xxStandard_Log_Format, e um Module_xxExtended_Log_Format que dizem respeito a cada módulo do *xxUniversal Log Format*.

5.4 *Interface* do Formato de Representação de Eventos

Nesta secção são apresentadas todas as decisões de arquitetura e implementação do sistema. No âmbito desta dissertação o limite para a exploração da arquitetura, tendo como base a anatomia de um SIEM está ilustrado na Figura 23, tendo como foco a fronteira da correlação, não sendo esta componente abordada.

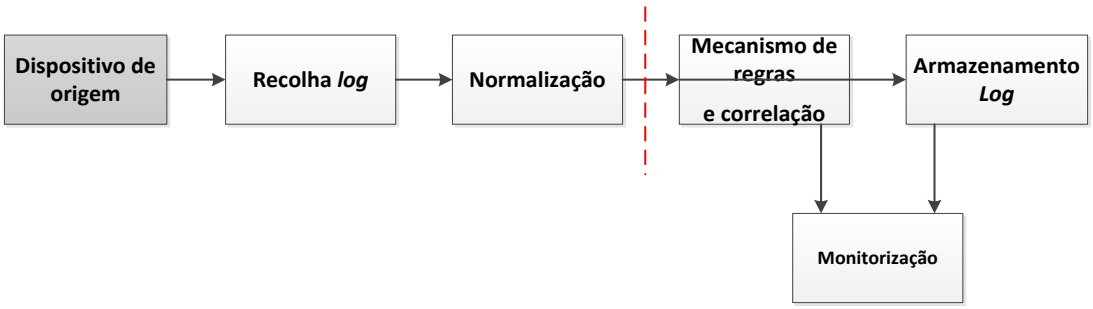


Figura 23 - Fronteira da exploração da arquitetura (Adaptado:(Miller & Pearson, 2011))

5.4.1 Arquitetura da interface

A proposta deste sistema é composta por um conjunto de mecanismos que tem como finalidade recolher analisar, tratar e normalizar os eventos provenientes de diversos ficheiros de *logs*, resultando num formato de representação único. As decisões latentes na especificação da arquitetura advêm das ilações retiradas nas análises efetuadas, tendo o objetivo de ir em encontro ao formato de representação proposto. Na Figura 24 é apresentada a proposta da arquitetura, que servirá de base para a implementação do sistema.

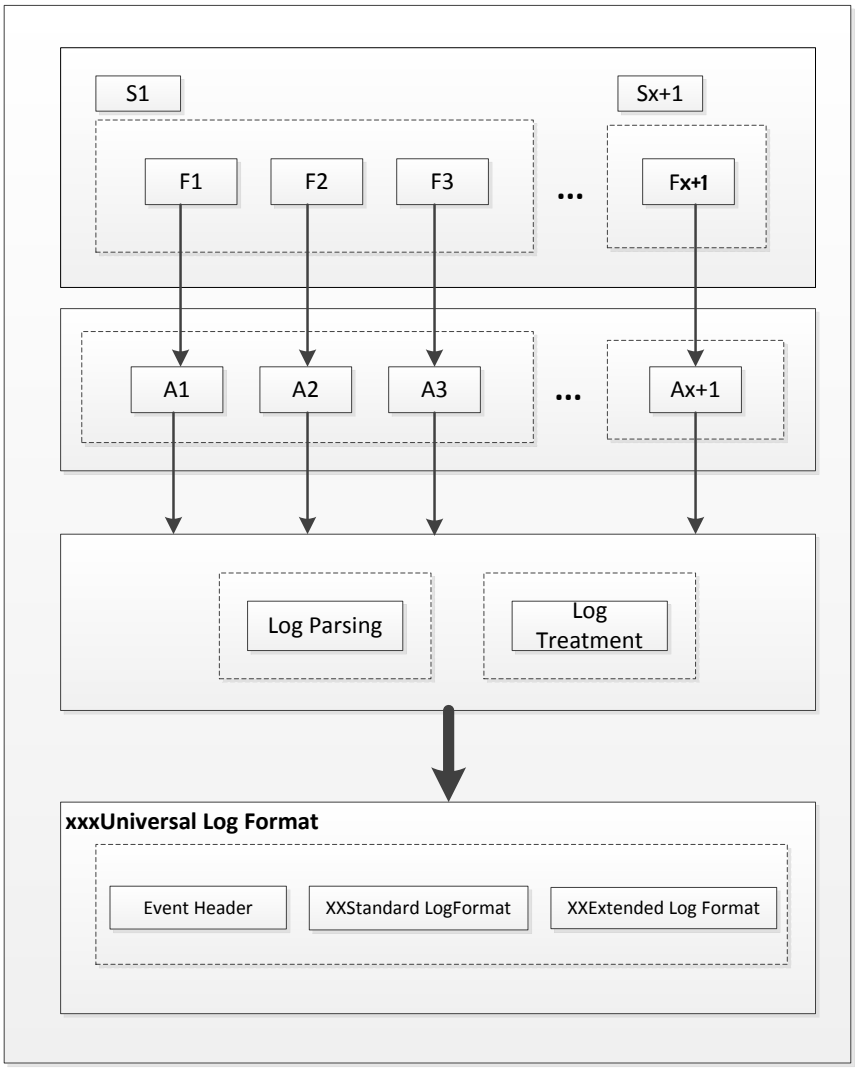


Figura 24 - Arquitetura da interface

Como representado na arquitetura, os eventos advém de diversas origens e cada sensor (Sx) gera *logs* num formato (F) ou em diversos formatos (Fx+1), cabendo a um agente (A) ou aos vários agentes (Ax+1) a responsabilidade de recolha. Devido à heterogeneidade dos formatos, a cada agente deve ser atribuído um ficheiro de *log* específico, que trata de forma particular cada sensor. Uma vez recolhidos os *logs* é efetuada a transformação para obter o formato normalizado. O objetivo do tratamento não é simplesmente de o transformar um ficheiro *log*, mas sim torná-lo mais fácil na partilha de informação e simplificar qualquer processamento adicional. Para o processo de transformação é necessário ter especificado o formato de representação do evento, bem como as especificações do formato *xxUniversal Log Format*. Qualquer *log* pode ser convertido para este formato, de tal forma que todos os registos possam ser analisados por qualquer sistema.

5.4.2 Implementação da interface

Apresentada a arquitetura, segue a descrição da sua implementação. O sistema em desenvolvimento tem por base *scripts* em *python* e ficheiros na linguagem XML para as configurações e especificações necessárias. Cada *script* diz respeito a uma classe e é responsável por recolher e transformar o *log* no formato de representação *xxUniversal Log Format.* Nos *scripts* são também utilizadas *Regular Expressions* de forma a extrair a informação para o evento normalizado. As *Regular Expressions* estão incorporadas dentro do *python* e permitem especificar um conjunto de regras e sequências de modo a facilitar a procura e processamento padrões.

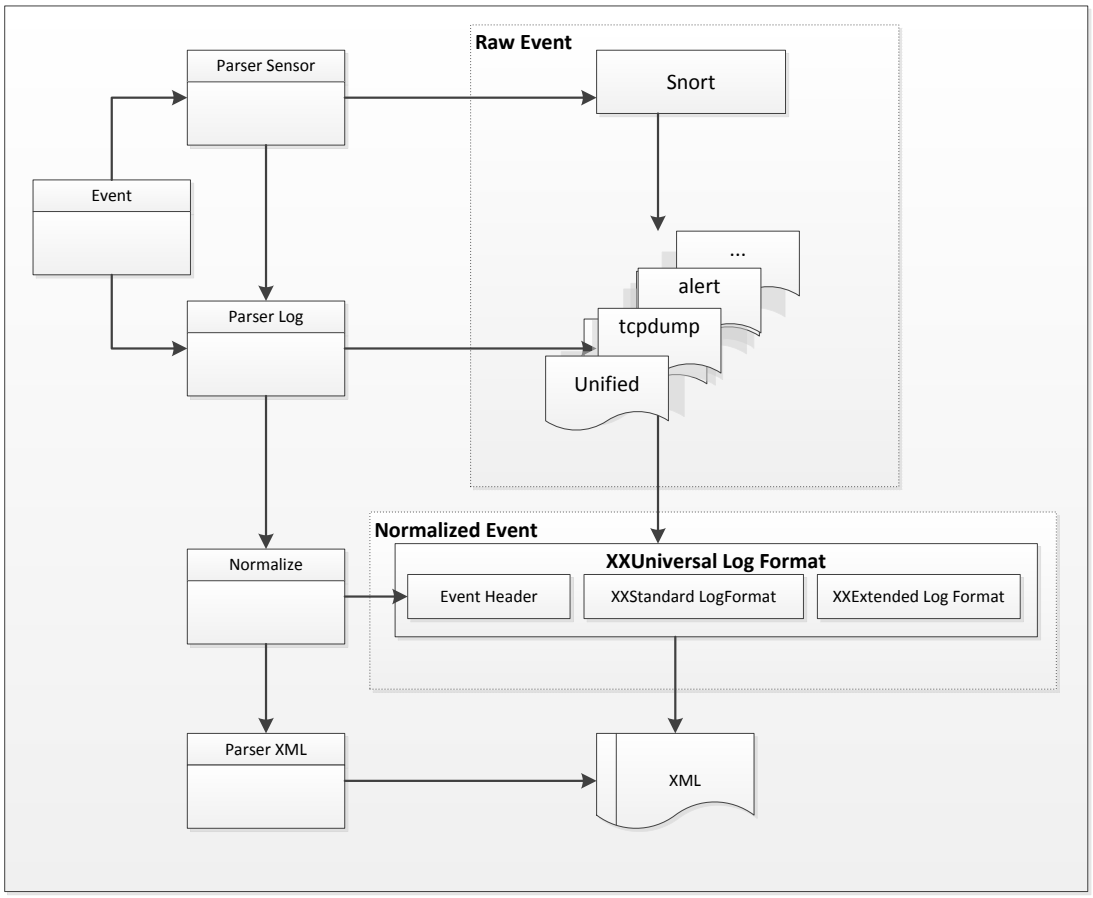


Figura 25- Implementação exemplo do sensor Snort

Como se pode verificar na Figura 25, o exemplo escolhido para ilustrar a implementação do sistema incidiu no sensor Snort. De seguida são descritas as classes que compõem o sistema:

- Classe *Parser Sensor*

A classe *Sensor* contém toda a informação sobre o sensor. Como cada sensor possuí a sua própria estrutura e os seus formatos de representação, é necessário implementar funções que caracterizem esse sensor. Por exemplo, nesta classe estão especificados todos os formatos e toda a informação considerada relevante para caracterizar esse sensor.

- Classe Event

Na classe Event estão definidos todos os tipos de classes identificadas (IDS, HIDS, ARP, entre outros) por grupos de informação (*Network Information*, *System Information* e *Aplication Information*) e sensores, bem como os atributos que o compõem.

- Classe *Parser Log*

Identificado o sensor e o formato utilizado é necessário fazer um processo de extração e tratamento do ficheiro, para que posteriormente se possa utilizar essa informação. Esta classe pode ser caracterizada como um pré-processamento do formato, em que transforma a informação em linguagem percetível. Nesta classe são utilizadas as *Regular Expressions* através do módulo "re" e são importadas as Classes *Event* e *Parser Sensor*. Para ilustrar as características desta classe, na Figura 26 está presente um evento retirado do ficheiro *Alert* do Snort, sem qualquer tipo de tratamento.

```
07/01-17:02:43.524564  [**] [1:1411:10] SNMP public access udp [**]
[Classification: Attempted Information Leak] [Priority: 2] {UDP}
193.137.8.8:60919 -> 192.168.233.41:161
```

Figura 26 - Evento Raw Snort

Através das especificações definidas para este tipo de formato é possível realizar um pré-processamento do evento, onde é produzido uma nova versão do ficheiro de *log*. Os atributos são delimitados pelo caracter ";" e os atributos são separados do nome como se pode verificar na Figura 27.

07/01-17:02:43; 1; 1001; SNMP public access udp ; Attempted Information Leak; 2; UDP; 193.137.8.8; 60919; 192.168.233.41; 161;

Figura 27- Evento pré-processado

Neste evento pré-processado já está identificado o grupo que se insere. Neste cenário o grupo é o *Network Information* (1) e a classe IDS (1001).

- Classe *Normalize*

Na Classe *Normalize* são implementadas todas as funções que definem o formato de representação proposto. Está classe é responsável pela construção do formato *xxUniversal Log Format* e é a principal classe da arquitetura. Nesta etapa, de acordo com a granularidade inerente ao evento (Group: Network Information -> Class: IDS) é aplicado um conjunto de ações de modo a ir em encontro do formato de representação. A Figura 28 ilustra o tipo de operação realizada, a partir do exemplo anterior.

|E0001|07/01-17:02:43|1|1001| 193.137.8.8 192.168.233.41 60919 161 UDP ^2^Attempted Information Leak^SNMP public access udp^

Figura 28 - Evento normalizado do formato xxUniversal Log Format

A utilização de um ficheiro único de registo para todos os grupos de informação é a estratégia mais simples e permite identificar as causas de falhas relacionadas com a intercalação de eventos gerados pelos vários componentes. No entanto, a utilização de um ficheiro de registo único para todo o sistema resulta na necessidade de um modelo com grandes capacidades de processamento. A estratégia escolhida foi a utilização de ficheiros de registos focando os eventos gerados pelas componentes individuais, reduzindo assim o tamanho do espaço de execução e os resultados em modelos que representam precisamente os eventos gerados pela execução de componentes.

- Classe Parser XML

A classe *Parser XML* traduz o formato *xxUniversal Log Format* para a linguagem de representação escolhida (XML).

```
<?xml version="1.0" encoding="UTF-8"?>
- <Format Type="xxUniversal Log Format">
  - <Group Type="Network Information">
    - <Class Type="IDS">
      - <Module_Header>
        - <Attribute>
            <name>E0001</name>
            <Description>Identifier of the event</Description>
          </Attribute>
        - <Attribute>
            <name>07/01-17:02:43</name>
            <Description>Time at which the event occurred</Description>
          </Attribute>
        - <Attribute>
            <name>1</name>
            <Description>Group Identification</Description>
          </Attribute>
        - <Attribute>
            <name>1001</name>
            <Description>Class Identification</Description>
          </Attribute>
        - <Attribute>
            <name>192.168.233.2</name>
            <Description>Identifier of the sensor responsible for the generation of the event</Description>
          </Attribute>
        </Module_Header>
      - <Module_xxStandard_Log_Format>
        - <Attribute>
            <name>193.137.8.8</name>
            <Description>The source IP address within the event</Description>
          </Attribute>
        - <Attribute>
            <name>192.168.233.41</name>
            <Description>The Destination_IP address within the event</Description>
          </Attribute>
        - <Attribute>
            <name>60919</name>
            <Description>The source port within the event</Description>
          </Attribute>
```

Figura 29 - Representação parcial do evento em XML

A Figura 29 mostra o resultado do formato de representação proposto, com base no exemplo anterior, na linguagem XML.

6. Conclusões e Trabalho Futuro

O presente capítulo contém uma reflexão sobre o trabalho realizado ao longo desta dissertação e uma análise crítica do contributo efetuado em torno dos formatos de representação de eventos, bem como as suas limitações. Por fim é descrito o trabalho a realizar num futuro próximo de forma a acrescentar valor ao formato de representação proposto bem como à *interface* desenvolvida

6.1 Conclusões

Face à revisão da literatura efetuada no âmbito dos formatos de representação de eventos de segurança da informação, é possível contastar que a utilização dos eventos tem bastante relevância para a segurança da informação. Contudo, esta técnica não tem sido muito utilizada, em grande parte devido à sua complexidade, contribuído muito para isso, o facto dos formatos de representação dos *log* de eventos não apresentarem uma sintaxe e semântica comum. Perante este facto, algumas organizações e instituições têm demonstrado o seu interesse na área com desenvolvimentos e pesquisas em torno de eventos com formatos e expressões comuns. Na área da investigação, apesar de não se focarem exclusivamente no problema, a comunidade científica tem contribuído com algumas abordagens na correlação de eventos, análise forense, gestão de *logs*, auditoria, monitorização, entre outros. A nível organizacional existem várias tentativas para desenvolver *standards* de interoperabilidade nos *logs e eventos*. Algumas tentativas não conseguiram ter o impacto ou importância esperada pelo motivo de serem proprietárias, ou seja não é possível modificar e experimentar com rigor, ou simplesmente por estarem descontinuadas.

O trabalho desta dissertação é um contributo importante para colmatar a lacuna existente na formalização da representação dos eventos de segurança de informação, pois de uma forma estruturada e modular permite traduzir os eventos, independentemente da sua origem e formato, para uma linguagem comum. Este facto, permite que os eventos de segurança possam ser utilizados como fonte de informação

útil e legível, tornando-se uma mais-valia, por exemplo, na interpretação dos eventos por parte dos responsáveis pela segurança de informação, ou aquando a necessidade de correlação de eventos, ou ainda do seu processamento automático.

6.2 Análise Crítica

Concluído o trabalho, é possível retirar algumas limitações ao contributo que é proposto. No que diz respeito ao formato de representação de eventos de segurança, uma das principais limitações resultou do facto da ferramenta OSSIM ser principalmente vocacionada para dados relacionados com a rede. Este facto, apesar de não ter afetado o grupo *Network Information* e *System Information*, dificultou substancialmente a definição do grupo *Application Information*. Apesar de lógica, a criação deste grupo evidencia algumas lacunas de informação existentes em torno deste domínio de informação.

Outra limitação identificada prende-se com os critérios de tomada de decisão para a especificação dos atributos do formato. Foram utilizados vários critérios, contudo para uma maior consolidação do formato proposto é interessante utilizar outras abordagens, nomeadamente a criação de vistas. Por exemplo, uma possível vista de exploração está relacionado com a granularidade do evento, ou seja, com o nível de detalhe que cada atributo pode oferecer. A exploração desta e outras vistas adicionais não foram utilizadas no âmbito da presente dissertação e serão abordadas no futuro. Também seria importante a utilização de algoritmos para a validação dos atributos pré-definidos da componente *xxStandard Log Format*, pois é a componente mais importante do formato. Contudo, todos as componentes devem ser alvo de validação com base nesses algoritmos de modo a avaliar a eficácia do formato.

No que diz respeito ao modelo, a principal limitação foi a não validação do modelo em ambiente de testes, com base em, por exemplo, cenários de ataques. É importante formalizar o formato em termos de comportamento do automatismo proposto na arquitetura, pois apesar de funcional, apenas foram realizados testes dos *scripts* de forma individual. Deste modo, apesar de o modelo permitir validar a viabilidade do formato de representação proposto, não é possível ter a perceção da eficácia e desempenho do modelo.

6.3 Trabalho Futuro

Futuramente existe um conjunto de aspetos que podem ser explorados de forma a acrescentar maior valor ao projeto apresentado, muitas das quais as enunciadas anteriormente.

Posto isto, para trabalho futuro será relevante:

- Devido à complexidade subjacente aos *logs* do *Windows* será interessante explorá-los de forma individual e verificar a possibilidade de integração do contributo *CEE-Enhanced Syslog*;
- Valorizar e aumentar a informação do grupo *Application Information;*
- Consolidação do formato com a utilização de outras vistas como fator de decisão na especificação do formato;
- Utilizar algoritmos que validem os atributos do formato, de modo a avaliar a eficácia do formato de representação;
- Automatizar o sistema proposto;
- Validar o formato e respetiva *interface*, com base em dados reais, gerados em um ambiente controlado;
- Criar um módulo capaz de correlacionar eventos de modo a filtrar e a otimizar a informação dos eventos;
- Estruturar algoritmos de forma a acrescentar inteligência aos eventos de segurança.

É também ambição e objetivo futuro para validação do formato efetuar uma validação por pares, de forma a aumentar a credibilidade e a qualidade do formato de representação proposto.

Esta página foi colocada propositadamente em branco.

Referências Bibliográficas

A. R. Hevner, S. T. March, J. Park, and S. R. (2004). Design science in information systems research. *MIS Quarterly*, 75–105.

Amiri, F., Gharaee, H., & Enayati, A. R. (2011). A complete operational architecture of alert correlation. *2011 International Conference on Computational Aspects of Social Networks (CASoN)*, 243–248. doi:10.1109/CASON.2011.6085952

ArcSight. (2010). *Common Event Format*.

Bray, R., Cid, D., & Hay, A. (2008a). *OSSEC host-based intrusion detection guide*. Syngress Publishing, Inc.

Bray, R., Cid, D., & Hay, A. (2008b). Working With Rules. In *OSSEC host-based intrusion detection guide* (p. 97).

Calder, A. (2006). *Information security based on ISO 27001/ISO 17799: a management guide*. (V. H. Publishing, Ed.).

Chen, W., & Yeung, D. (2006). Defending against TCP SYN flooding attacks under different types of IP spoofing. Conference on Mobile Communications and Learning Technologies. IEEE.

Chuvakin, A., Schmidt, K., & Phillips, C. (2012). *Logging and Log Management: The Authoritative Guide to Dealing with Syslog, Audit Logs, Events, Alerts and other IT "Noise." Logging and Log Management: The Authoritative Guide to Dealing with Syslog, Audit Logs, Events, Alerts and Other It Noise* (p. 413).

Colace, F., De Santo, M., & Ferrandino, S. (2012). A Slow Intelligent Approach for the Improvement of Intrusion Detection and Prevention System. *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 130–137.

Coviello, A. (2011). Open letter to RSA customers. *RSA [database Online]*.

Cuppens, F. (2001). Managing alerts in a multi-intrusion detection environment. *Seventeenth Annual Computer Security Applications Conference*, 22–31.

Danyliw, R., Meijer, J., & Demchenko, Y. (2007). The incident object description exchange format. Retrieved from http://www.ietf.org/rfc/rfc5070.txt

Deokar, B., & Hazarnis, A. (2012). Intrusion Detection System using Log Files and Reinforcement Learning. *International Journal of Computer Applications*, *45*(19), 28–35.

Elgin, B., Lawrence, D., & Riley, M. (2012). Coke Gets Hacked And Doesn't Tell Anyone. *Bloomberg News*.

Fooprateepsiri, R., & Kurutach, W. (2010). A Highly Robust Approach Image Identification based-on Hausdorff- Trace Transform. International Journal of Digital Content Technology and its Applications. 4(1).

Fry, C., & Nystrom, M. (2009). Security Monitoring. In 2009 O'Reilly Media, Inc. (Ed.), (p. 256).

Gerhards, R. (2009). The syslog protocol. Retrieved from http://tools.ietf.org/html/rfc5424#section-6.1

Gerhards, R. (2012). CEE-enhanced syslog defined. Retrieved January 16, 2013, from http://blog.gerhards.net/2012/03/cee-enhanced-syslog-defined.html

Google. (2012). The Protocol Buffers: Developer Guide. Retrieved from https://developers.google.com/protocol-buffers/docs/overview

GU, Z., & Li, Y. (2011). Research of Security Event Correlation based on Attribute Similarity. *JDCTA: International Journal of Digital Contentent Technology and Its Applications*, *5*(6), 222–228.

Guimarães, N., & Marques, A. (1992). Projecto e Implementação de um Sistema de Data Webhousing. *Di.uminho.pt*. Retrieved from http://www.di.uminho.pt/~prh/uce15-0809/g14.pdf

Hammoud, N. (2009). Decentralized log event correlation architecture. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems - MEDES '09* (p. 480). New York, New York, USA: ACM Press.

Heinbockel, W., Judge, J., McQuaid, R., Chuvakin, A., & Marty, R. (2008). *Common Event Expression* (p. 30).

Hernandez, J. (2010). *Security Information and Event Management: Business Benefits and Security, Governance and Assurance Perspectives* (p. 12). Retrieved from www.isaca.org/siem

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 75–105.

Houle, K., Weaver, G., Long, N., & Thomas, R. (2001). Trends in denial of service attack technology. *CERT Coordination Center*, (October), 0–20.

Howard, J. D., & Longstaff, T. A. (1998). A Common Language for Computer Security Incidents, (October).

ISO/IEC. (2009). ISO/IEC 27004:2009, Information technology — Security techniques — Information security management — Measurement. Geneva, Switzerland: International Commission Organization for Standardization/International Electrotechnical.

Jiang, Z., & Hassan, A. (2008). An automated approach for abstracting execution logs to execution events. *Journal of Software Maintenance and Evolution: Research and Practice*, 249–267.

Jingxin, W., & Zhiying, W. (2007). Security Event Management System based on Mobile Agent Technology. *2007 IEEE Intelligence and Security Informatics*, 166–171.

Jones, B. (2010). *Understanding and Selecting SIEM / Log Management* (p. 40). Retrieved from https://securosis.com/assets/library/reports/Securosis_Understanding_Selecting_SIEM_LM_FINAL.pdf

Kahn, C. ord, Porras, P., Staniford-Chen, S., & Tung, B. (1998). A Common Intrusion Detection Framework. Citeseer, 0–17.

Karlzén, H. (2009). *An Analysis of Security Information and Event Management Systems-The Use or SIEMs for Log Collection, Management and Analysis*. Retrieved from http://publications.lib.chalmers.se/publication/89572

Kent, K., & Souppaya, M. (2006). Guide to computer security log management. *NIST Special Publication*. Retrieved from http://logrhythm.com/Portals/0/resources/NIST Guide Log Mgmt SP800-92.pdf

Lambert, P. L. (2012). Analysis of a targeted cyber attack.

Li, L. (2010). Research on the network security management based on data mining. *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, V5–184–V5–187.

Lin, C., Zhitang, L., & Cuixia, G. (2009). Automated Analysis of Multi-Source Logs for Network Forensics. *2009 First International Workshop on Education Technology and Computer Science*, 660–664.

Madani, A., Rezayi, S., & Gharaee, H. (2011). Log management comprehensive architecture in Security Operation Center (SOC). *2011 International Conference on Computational Aspects of Social Networks (CASoN)*, 284–289.

McAfee, I. (2010). How To Respond To The Recent Microsoft Internet Explorer.

Miller, D., & Pearson, B. (2011). *Security information and event management (SIEM) implementation*. Retrieved from http://media.matthewsbooks.com.s3.amazonaws.com/documents/tocwork/007/9780071701099.pdf

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review. Retrieved from http://dl.acm.org/citation.cfm?id=997150.997156

MITRE. (2011). Common Event Expression. Retrieved from http://cee.mitre.org/language/0.6/CEE_Architecture_Overview-v0.6.pdf

Morin, B., Mé, L., Debar, H., & Ducassé, M. (2009). A logic-based model to support alert correlation in intrusion detection. *Information Fusion*, *10*(4), 285–299.

Murray, T. (2003). *Getting a Handle on Security Events - GSEC Practical Assignment*. Retrieved from http://www.giac.org/paper/gsec/2931/handle-security-events/104928

Myers, J., Grimaila, M. R., & Mills, R. F. (2011). Log-Based Distributed Security Event Detection Using Simple Event Correlator. *2011 44th Hawaii International Conference on System Sciences*, 1–7.

Nawyn, K. (2003). A Security Analysis of System Event Logging with Syslog. *SANS Institute. Citeseer*.

Notícias ao Minuto. (2012). Informática Computadores do Governo alvo de 800 mil ataques em Agosto. Retrieved from http://www.noticiasaominuto.com/pais/11134/computadores-do-governo-alvo-de-800-mil-ataques-em-agosto#.UPnmxB1LNfQ

Ogle, D., Kreger, H., & Salahshour, A. (2004). Canonical situation data format: the common base event V1. 0.1. *IBM Corporation*. Retrieved from http://www.eclipse.org/tptp/platform/documents/resources/cbe101spec/CommonBaseEvent_SituationData_V1.0.1.pdf

Salama, S. E., I. Marie, M., El-Fangary, L. M., & K. Helmy, Y. (2011). Web Server Logs Preprocessing for Web Intrusion Detection. *Computer and Information Science*, *4*(4), 123–133.

Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS) Recommendations of the National Institute of Standards and Technology. *800-94*.

Shrivastava, A. (2012). An Approach for Sytem Logs Analysis By Using Association Rule Mining. *International Journal of Research in Computer Engineering and Electronics*, 1–5.

SOL. (2012). PayPal já perdeu 4,7 milhões de euros com ataques do Anonymous. Retrieved from http://sol.sapo.pt/inicio/Tecnologia/Interior.aspx?content_id=63469

Stewart, J., Chapple, M., & Gibson, D. (2012). *CISSP: Certified Information Systems Security Professional Study Guide*. John Wiley & Sons, Inc. Sixth Ed.

Susanto, H., Almunawar, M., & Tuan, Y. (2011). Information security management system standards: A comparative study of the big five. *International Journal of Electrical & Computer Sciences*.

Vaishnavi, V., & Kuechler, B. (2004). Design Science Research in Information Systems. *Association For Information Systems*.

Whitman, M., & Mattord, H. (2009). *Principles of information security*. (2009 Cengage Learning EMEA, Ed.) (Third Edit., p. 598). Boston.

Wilshusen, G., & Powner, D. (2009). CYBERSECURITY: Continued Efforts Are Needed to Protect Information Systems from Evolving Threats. Retrieved from http://www.gao.gov/new.items/d10230t.pdf

Zhaojun, G., Yong, L., & Wenjing, N. (2010). Analysis and implement of PIX firewall syslog log. *2010 2nd IEEE International Conference on Information Management and Engineering*, 185–189.

Esta página foi colocada propositadamente em branco.

Anexos

Anexo 1. Sensores por classes (ver http://goo.gl/ZgNmeK)

Class	Sensores																										
Admin System/Security	Drupal Wiki	imperva-securesphere	panda-as	webmin																							
Agentless Monitoring	sitescope																										
AntiMalware	panda-se																										
AntiSpam	mcafee-antispam	optnet	spamassasin																								
Antivirus	Avast	Clamav	Gfi	mcafee	sophos	symanteo-amc																					
Authentication	pam-unix	rsa-secureid	sudo	token-rsa																							
Camera Video	Motion																										
CC transaction	vplus																										
Central Policy	mcafee-epo																										
Cluster Service	Clurgmgr	HeartBeat	serviceguard																								
Correlation	post-correlation																										
Data Loss Prevention	Fidelis																										
Fingerprint	p0f	p0f_eth1																									
Firewall	Cisco Asa Firewall	Cisco Firewall	Cisco PIX	Cyberguard	Fortigate	fwl-alt	fwlngr60	ipfw	lucent-brick	Motorola-firewall	netkeeper-fw	netscreen-firewall	netscreen-igs	netscreen-manager	paloalto	sidewinder	sonicwall	stonegate									
HIDS	osiris	ossec	ossec-idm	ossim-agent	ossim-monitor																						
Honeypot	amun-honeypot	Artemisa	Dionea	GlastopfNG	Honeyid	mwcollected	nepenthes																				
Host Malware	malwaredomainlist-monitor																										
IDM	snare																										
IDS	Bro	Cisco IDS	intrushield	Kismet	modsecurity	netkeeper-nids	snort_syslog																				
IPS	Cisco IPS Syslog	Dragon	radware-ips	siteprotector	stonegate-ips	tippingpoint																					
mail	Axigen-mail	Exchange	ironport	postfix	sendmail																						
Management Software	sap																										
Messaging Security	trendmicro																										
Mobile	iphone																										
Monitor Arp	Arpalert	Arpwatch																									
Monitoring Network	nagios	realsecure	rrd																								
Monitoring windows	wmi-application-logger	wmi-security-logger	wmi-security-logger-srv2008	wmi-system-logger																							
Packet Filter	pf																										
PADS	pads	pads_eth1																									
Port Scanner	nmap																										
Process	Elsfe																										
Security Plataform	Bit9																										
Service Server	Apache	Bind	DHCP	Dovecot	IIS	Linuxdhcp	nfs	openldap	oracle-sql	oracle-syslog	pureftpd	radiator	shrubbery-tacacs	smbd	ssh	ssh-remote	syslog	tacacs-plus	vandyke-vshell	vsftpd								
Storage Management	HP-Eva																										
System Event	aix-audit	cisco-nexus-nx-os	raslogd																								
Traffic Devices	Allot	Alteon OS	Aruba	Ascenlink	Cisco-ace	Cisco ACS	Cisco ASR	Cisco-Router	Cisco WLC	citrix-netscaller	Enterasys-Pimatrix	Extreme-switch	extreme-wireless	F5	Juniper-SRX	netgear	nortel-switch	proxim-orinoco									
Usb Control	usbudev																										
Virtualization	vmware-esxi	vmware-vcenter	vmware-vcenter-sql	vmware-workstation																							
VPN	Cisco-3030 VN	Cisco VPN	F5 Firepass Network	isa	Juniper-vpn																						
Vulnerability Scanner	nessus	nessus-detector	nessus-monitor																								
WAF	Airlock																										
Web Aplication	moodle																										
Web Proxy	squid																										
Web Security	Alladin	Bluecoat	Fortiguard	websense																							

Anexo 2. Formato de Representação em XML

Group System Information

```
<?xml version='1.0' encoding='UTF-8' ?>
<Format Type = "xxUniversal Log Format">
    <Group Type =  "System Information">
        <Class Type = "HIDS">
            <Module_Header>
                    <Attribute>
                        <name>EventID</name>
                        <Description>Identifier of the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Timestamp</name>
                        <Description>Time at which the event occurred</Description>
                    </Attribute>
                    <Attribute>
                        <name>GID</name>
                        <Description>Group Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>CID</name>
                        <Description>Class Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>Sid_Event</name>
                        <Description>Identifier of the sensor responsible for the generation
                        of the event</Description>
                    </Attribute>
            </Module_Header>
            <Module_xxStandard_Log_Format>
                    <Attribute>
                        <name>Hostname</name>
                        <Description>Unique Identifier source of the host associated to an
                        event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Source_IP</name>
                        <Description>The source IP address within the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Destination_IP</name>
                        <Description>The Destination_IP address within the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Source_Port</name>
                        <Description>The source port within the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Source_User</name>
                        <Description>User who has been identified as the event generator
                        </Description>
                    </Attribute>
                    <Attribute>
                        <name>Destination_User</name>
                        <Description>Extracts the destination (target) username</Description>
                    </Attribute>
                    <Attribute>
                        <name>Service Name</name>
                        <Description>The service within the event</Description>
```

```
        </Attribute>
        <Attribute>
            <name>Process</name>
            <Description>The process within the event</Description>
        </Attribute>
        <Attribute>
            <name>System Name</name>
            <Description>The system name within the event (linux, windows, mac
            OS, etc)</Description>
        </Attribute>
        <Attribute>
            <name>Message</name>
            <Description>The message section of the event.</Description>
        </Attribute>
        <Attribute>
            <name>Ext_Reference</name>
            <Description>External References information (CVE, OSVDB, Secunia,
            etc) that is included with the event</Description>
        </Attribute>
</Module_xxStandard_Log_Format>
<Module_xxExtended_Log_Format>
        <Attribute>
            <name>Action</name>
            <Description>The action taken within the event</Description>
        </Attribute>
        <Attribute>
            <name>Location</name>
        <Description>Where the log came from</Description>
        </Attribute>
        <Attribute>
            <name>ID</name>
            <Description>Any ID decoded as the ID from the event</Description>
        </Attribute>
        <Attribute>
            <name>Status</name>
            <Description>The decoded status within the event</Description>
        </Attribute>
        <Attribute>
            <name>Command</name>
            <Description>The command being called within the event</Description>
        </Attribute>
        <Attribute>
            <name>URL</name>
            <Description>The URL information within the event</Description>
        </Attribute>
        <Attribute>
            <name>Domain</name>
            <Description>The Domain of the system within the event</Description>
        </Attribute>
        <Attribute>
            <name>Data</name>
            <Description>Any additional data that you want to extract from the
            payload of the event</Description>
        </Attribute>
        <Attribute>
            <name>Full_Log</name>
            <Description>The entire event</Description>
```

```
</Attribute>
<Attribute>
    <name>Destination_Port</name>
    <Description>The destination port within the event</Description>
</Attribute>
<Attribute>
    <name>Protocol</name>
    <Description>The protocol within the event</Description>
</Attribute>
<Attribute>
    <name>Filename</name>
    <Description>File used in an event</Description>
</Attribute>
<Attribute>
    <name>Size</name>
    <Description>Size of the object associated with the event
    </Description>
</Attribute>
<Attribute>
    <name>UserAgent</name>
    <Description>The user agent within the event</Description>
</Attribute>
<Attribute>
    <name>Type</name>
    <Description>Is a description of the event in order to understand
    what the event is about</Description>
</Attribute>
<Attribute>
    <name>Method</name>
    <Description>The method within the event</Description>
</Attribute>
<Attribute>
    <name>TTY</name>
    <Description>Teletype terminal associated with the event</Description>
</Attribute>
<Attribute>
    <name>Path</name>
    <Description>Path of file within the event</Description>
</Attribute>
<Attribute>
    <name>Petition</name>
    <Description>The petition within the event</Description>
</Attribute>
<Attribute>
    <name>Target</name>
    <Description>The target within the event</Description>
</Attribute>
<Attribute>
    <name>Version</name>
    <Description>The version of system within the event</Description>
</Attribute>
<Attribute>
    <name>http_Request</name>
    <Description>The Hypertext Transfer Protocol within the event
    </Description>
</Attribute>
<Attribute>
```

```
    <name>Agent_Name</name>
    <Description>The OSSEC agent name within the event(example:
    host1,host2)</Description>
</Attribute>
<Attribute>
    <name>Sudo</name>
    <Description>The sudo used in the event</Description>
</Attribute>
<Attribute>
    <name>Password</name>
    <Description>The Password used in an event</Description>
</Attribute>
<Attribute>
    <name>SSHD</name>
    <Description>Name of the Solid State Hybrid Drive</Description>
</Attribute>
<Attribute>
    <name>Dst_Host</name>
    <Description>Destination Host associated with the event</Description>
</Attribute>
<Attribute>
    <name>New_md5sum</name>
    <Description>The new MD5 hash within the event</Description>
</Attribute>
<Attribute>
    <name>Old_md5sum</name>
    <Description>The new MD5 hash within the event</Description>
</Attribute>
<Attribute>
    <name>New_sha1sum</name>
    <Description>The new SHA-1 hashes within the event</Description>
</Attribute>
<Attribute>
    <name>Old_sha1sum</name>
    <Description>The old SHA-1 hashes within the event</Description>
</Attribute>
<Attribute>
    <name>Package</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Logon_ID</name>
    <Description>Logon ID value within the event</Description>
</Attribute>
<Attribute>
    <name>Win_Event_Name</name>
    <Description>Windows event name associated with the event
    </Description>
</Attribute>
<Attribute>
    <name>http_Version</name>
    <Description>Version of the Hypertext Transfer Protocol</Description>
</Attribute>
<Attribute>
    <name>Response_Code</name>
    <Description>The Hypertext Transfer Protocol response code within
    the event</Description>
```

```
            </Attribute>
            <Attribute>
                <name>http_Protocol</name>
                <Description>The Hypertext Transfer Protocol within the event
                </Description>
            </Attribute>
            <Attribute>
                <name>http_Code</name>
                <Description>The Hypertext Transfer Protocol code associated with
                the event </Description>
            </Attribute>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Honeypot">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Vulnerability_Scanner ">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
```

```
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Virtualization">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "System Event">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
```

```
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Process">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Monitoring_windows">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
```

```
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Mobile">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Fingerprint">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
```

```
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Cluster_Service">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Central_Policy">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
```

```
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Credit_Card_Transaction">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Authentication ">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
```

```
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Agentless_Monitoring ">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
            </Attribute>
            <Attribute>
                <name>Sid_Event</name>
                <Description>Identifier of the sensor responsible for the generation
                of the event</Description>
            </Attribute>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Service_Server">
    <Module_Header>
        <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
```

```
                    </Attribute>
                    <Attribute>
                        <name>GID</name>
                        <Description>Group Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>CID</name>
                        <Description>Class Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>Sid_Event</name>
                        <Description>Identifier of the sensor responsible for the generation
                        of the event</Description>
                    </Attribute>
                </Module_Header>
                <Module_xxStandard_Log_Format>
                </Module_xxStandard_Log_Format>
                <Module_xxExtended_Log_Format>
                </Module_xxExtended_Log_Format>
            </Class>
            <Class Type = "Admin_System/Security ">
                <Module_Header>
                    <Attribute>
                            <name>EventID</name>
                            <Description>Identifier of the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Timestamp</name>
                        <Description>Time at which the event occurred</Description>
                    </Attribute>
                    <Attribute>
                        <name>GID</name>
                        <Description>Group Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>CID</name>
                        <Description>Class Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>Sid_Event</name>
                        <Description>Identifier of the sensor responsible for the generation
                        of the event</Description>
                    </Attribute>
                </Module_Header>
                <Module_xxStandard_Log_Format>
                </Module_xxStandard_Log_Format>
                <Module_xxExtended_Log_Format>
                </Module_xxExtended_Log_Format>
            </Class>
        </Group>
    </Format>
```

Group Application Information

```
<?xml version='1.0' encoding='UTF-8' ?>
<Format Type = "xxUniversal Log Format">
    <Group Type =  "Network Information">
        <Class Type = "Antivirus">
            <Module_Header>
                    <Attribute>
                        <name>EventID</name>
                        <Description>Identifier of the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Timestamp</name>
                        <Description>Time at which the event occurred</Description>
                    </Attribute>
                    <Attribute>
                        <name>GID</name>
                        <Description>Group Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>CID</name>
                        <Description>Class Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>Sid_Event</name>
                        <Description>Identifier of the sensor responsible for the generation
                        of the event</Description>
                    </Attribute>
            </Module_Header>
            <Module_xxStandard_Log_Format>
                    <Attribute>
                        <name>Source_IP</name>
                        <Description>The source IP address within the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Application</name>
                        <Description> </Description>
                    </Attribute>
                    <Attribute>
                        <name>Filename</name>
                        <Description>Source of Mac Address</Description>
                    </Attribute>
                    <Attribute>
                        <name>Path</name>
                        <Description>Destination of Mac Address</Description>
                    </Attribute>
                    <Attribute>
                        <name>Message</name>
                        <Description>The message section of the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Ext_Reference</name>
                        <Description>External References information (CVE, OSVDB, Secunia,
                        etc) that is included with the event</Description>
                    </Attribute>
            </Module_xxStandard_Log_Format>
            <Module_xxExtended_Log_Format>
                    <Attribute>
                        <name>Username</name>
```

```
    <Description></Description>
</Attribute>
<Attribute>
    <name>Action</name>
    <Description>The action taken within the event</Description>
</Attribute>
<Attribute>
    <name>Severity</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Reason</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Hostname</name>
<Description> </Description>
</Attribute>
<Attribute>
    <name>Virus</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Generator</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Sender_Address</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Receptor_Address</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Subject</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>What</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>File</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Mailer_ID</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Threat_Level</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Exploit</name>
    <Description>This is the TCP-header checksum of the packet (for
```

```
                checking packets integrity)</Description>
            </Attribute>
            <Attribute>
                <name>Risk_Name</name>
                <Description></Description>
            </Attribute>
            <Attribute>
                <name>Register</name>
                <Description></Description>
            </Attribute>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "PADS">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Web Security">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Web_Proxy">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Web_Aplication">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Windows_Aplication_Firewall">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Usb_Control">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
```

```
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Storage_Management">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Messaging_security">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Management_Software">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Host_Malware">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Anti_Malware">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Mail">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "Correlation">
    <Module_Header>
    </Module_Header>
    <Module_xxStandard_Log_Format>
    </Module_xxStandard_Log_Format>
    <Module_xxExtended_Log_Format>
```

```
            </Module_xxExtended_Log_Format>
        </Class>
        <Class Type = "Data_Loss_Prevention">
            <Module_Header>
            </Module_Header>
            <Module_xxStandard_Log_Format>
            </Module_xxStandard_Log_Format>
            <Module_xxExtended_Log_Format>
            </Module_xxExtended_Log_Format>
        </Class>
        <Class Type = "AntiSpam">
            <Module_Header>
            </Module_Header>
            <Module_xxStandard_Log_Format>
            </Module_xxStandard_Log_Format>
            <Module_xxExtended_Log_Format>
            </Module_xxExtended_Log_Format>
        </Class>
    </Group>
</Format>
```

Group Network Information

```
<?xml version='1.0' encoding='UTF-8' ?>
<Format Type = "xxUniversal Log Format">
    <Group Type =  "Network Information">
        <Class Type = "IDS">
            <Module_Header>
                    <Attribute>
                        <name>EventID</name>
                        <Description>Identifier of the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Timestamp</name>
                        <Description>Time at which the event occurred</Description>
                    </Attribute>
                    <Attribute>
                        <name>GID</name>
                        <Description>Group Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>CID</name>
                        <Description>Class Identification</Description>
                    </Attribute>
                    <Attribute>
                        <name>Sid_Event</name>
                        <Description>Identifier of the sensor responsible for the generation
                        of the event</Description>
                    </Attribute>
            </Module_Header>
            <Module_xxStandard_Log_Format>
                    <Attribute>
                        <name>Source_IP</name>
                        <Description>The source IP address within the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Destination_IP</name>
                        <Description>The Destination_IP address within the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Source_Port</name>
                        <Description>The source port within the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Destination_Port</name>
                        <Description>The destination port within the event</Description>
                    </Attribute>
                    <Attribute>
                        <name>Protocol</name>
                        <Description>Types of protocols allowed</Description>
                    </Attribute>
                    <Attribute>
                        <name>Interface</name>
                        <Description>Network interface</Description>
                    </Attribute>
                    <Attribute>
                        <name>Mac_Source</name>
                        <Description>Source of Mac Address</Description>
                    </Attribute>
                    <Attribute>
```

```
            <name>Mac_Destination</name>
            <Description>Destination of Mac Address</Description>
        </Attribute>
        <Attribute>
            <name>Message</name>
            <Description>The message section of the event.</Description>
        </Attribute>
        <Attribute>
            <name>Ext_Reference</name>
            <Description>External References information (CVE, OSVDB, Secunia,
            etc) that is included with the event</Description>
        </Attribute>
</Module_xxStandard_Log_Format>
<Module_xxExtended_Log_Format>
        <Attribute>
            <name>Severity</name>
            <Description>The severity ratings associated with the event
            </Description>
        </Attribute>
        <Attribute>
            <name>Action</name>
            <Description>The action taken within the event</Description>
        </Attribute>
        <Attribute>
            <name>ICMP_Code</name>
            <Description>Indicates the reason for the loss of the datagram
            </Description>
        </Attribute>
        <Attribute>
        <Attribute>
            <name>Rule_Name</name>
            <Description>The name of the rule associated to the event
            </Description>
        </Attribute>
            <name>Signature_Id</name>
        <Description>Its the signature of the event</Description>
        </Attribute>
        <Attribute>
            <name>Class_Type</name>
            <Description>The classtype keyword is used to categorize a rule as
            detecting an attack that is part of a more general type of attack
            class. Snort provides a default set of attack classes that are used
            by the default set of rules it provides</Description>
        </Attribute>
        <Attribute>
            <name>Tcp_Flags</name>
            <Description>The TCP flags shows what the sending TCP entity wants
            the receiving TCP entity to do.</Description>
        </Attribute>
        <Attribute>
            <name>ToS</name>
            <Description>Type of service could specify a datagram priority and
            request a route for low-delay, high-throughput,or highly-reliable
            service.</Description>
        </Attribute>
        <Attribute>
            <name>Size_Packet</name>
```

```
    <Description>Length of the sampled packet</Description>
</Attribute>
<Attribute>
    <name>Hostname</name>
    <Description>A host is a computer that is connected to a network,
    usually the Internet or other TCP/IP network,such as a LAN. Each
    host on a TCP/IP network has a unique IP address, which is a unique
    numeric identifier.</Description>
</Attribute>
<Attribute>
    <name>TTL</name>
    <Description>Check the IP header's  time-to-live (TTL)field
    </Description>
</Attribute>
<Attribute>
    <name>Tcp_ACK</name>
    <Description>Look for a specific TCP header acknowledgement number.
    </Description>
</Attribute>
<Attribute>
    <name>Tcp_Sequence</name>
    <Description>This is the TCP sequence number</Description>
</Attribute>
<Attribute>
    <name>Type</name>
    <Description>Is a description of the event in order to understand
    what the event is about</Description>
</Attribute>
<Attribute>
    <name>Checksum</name>
    <Description>This is the TCP-header checksum of the packet (for
    checking packets integrity)</Description>
</Attribute>
<Attribute>
    <name>Window_Size</name>
    <Description>Is the maximum amount of received data, in bytes, that
    can be buffered at one time on the receiving side of a connection.
    </Description>
</Attribute>
<Attribute>
    <name>Reason</name>
    <Description>The reason of the event</Description>
</Attribute>
<Attribute>
    <name>Description</name>
    <Description>Additional description about the event</Description>
</Attribute>
<Attribute>
    <name>Categories</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Sub_Category</name>
    <Description></Description>
</Attribute>
<Attribute>
    <name>Result_status</name>
```

```
                <Description></Description>
            </Attribute>
            <Attribute>
                <name>Netsumbler_Version</name>
                <Description></Description>
            </Attribute>
            <Attribute>
                <name>Snort_Sid</name>
                <Description>Identification of the snort rules</Description>
            </Attribute>
            <Attribute>
                <name>Snort_Cid</name>
                <Description>Normalized listing of alert/signature classification
                </Description>
            </Attribute>
            <Attribute>
                <name>Crypt</name>
                <Description></Description>
            </Attribute>
            <Attribute>
                <name>SSID</name>
                <Description>The public name of a wireless network within the eve
                </Description>
            </Attribute>
            <Attribute>
                <name>BSSID</name>
                <Description>Identify Access Points and Their Clients (Access poi
                mac address)</Description>
            </Attribute>
            <Attribute>
                <name>Channel</name>
                <Description>Identify of the communication channel</Description>
            </Attribute>
            <Attribute>
                <name>Attack_Name</name>
                <Description>The name of the attack associated with the event
                </Description>
            </Attribute>
    </Module_xxExtended_Log_Format>
</Class>
<Class Type = "ARP">
    <Module_Header>
            <Attribute>
                <name>EventID</name>
                <Description>Identifier of the event</Description>
            </Attribute>
            <Attribute>
                <name>Timestamp</name>
                <Description>Time at which the event occurred</Description>
            </Attribute>
            <Attribute>
                <name>GID</name>
                <Description>Group Identification</Description>
            </Attribute>
            <Attribute>
                <name>CID</name>
                <Description>Class Identification</Description>
```

```
        </Attribute>
        <Attribute>
            <name>Sid_Event</name>
            <Description>Identifier of the sensor responsible for the generation
            of the event</Description>
        </Attribute>
</Module_Header>
<Module_xxStandard_Log_Format>
        <Attribute>
            <name>Source_IP</name>
            <Description>The source IP address within the event</Description>
        </Attribute>
        <Attribute>
            <name>Destination_IP</name>
            <Description>The Destination_IP address within the event</Description>
        </Attribute>
        <Attribute>
            <name>Source_Port</name>
            <Description>The source port within the event</Description>
        </Attribute>
        <Attribute>
            <name>Destination_Port</name>
            <Description>The destination port within the event</Description>
        </Attribute>
        <Attribute>
            <name>Protocol</name>
            <Description>Types of protocols allowed</Description>
        </Attribute>
        <Attribute>
            <name>Interface</name>
            <Description>Network interface</Description>
        </Attribute>
        <Attribute>
            <name>Mac_Source</name>
            <Description>Source of Mac Address</Description>
        </Attribute>
        <Attribute>
            <name>Mac_Destination</name>
            <Description>Destination of Mac Address</Description>
        </Attribute>
        <Attribute>
            <name>Message</name>
            <Description>The message section of the event.</Description>
        </Attribute>
        <Attribute>
            <name>Ext_Reference</name>
            <Description>External References information (CVE, OSVDB, Secunia,
            etc) that is included with the event</Description>
        </Attribute>
</Module_xxStandard_Log_Format>
<Module_xxExtended_Log_Format>
        <Attribute>
            <name>Hostname</name>
            <Description>Unique Identifier source of the host associated to an
            event</Description>
        </Attribute>
        <Attribute>
```

```
                <name>Vendor</name>
                <Description>Manufacturer name of the Ethernet card</Description>
            </Attribute>
            <Attribute>
                <name>Type</name>
                <Description>Is a description of the event in order to understand
                what the event is about</Description>
            </Attribute>
            <Attribute>
                <name>Mac_Source_Old</name>
                <Description>Its the old interface source associated with the event
                </Description>
            </Attribute>
            <Attribute>
                <name>Mac_Destination_Old</name>
                <Description>Its the old interface destination associated with the
                event </Description>
            </Attribute>
            <Attribute>
                <name>Previous_Timestamp</name>
                <Description>Its the old timestamp associated with the event (before
                the change)</Description>
            </Attribute>
            <Attribute>
                <name>Delta</name>
                <Description>Its the difference between previous timestamp and
                actual timestamp</Description>
            </Attribute>
        </Module_xxExtended_Log_Format>
    </Class>
    <Class Type = "IPS">
        <Module_Header>
        </Module_Header>
        <Module_xxStandard_Log_Format>
        </Module_xxStandard_Log_Format>
        <Module_xxExtended_Log_Format>
        </Module_xxExtended_Log_Format>
    </Class>
    <Class Type = "Traffic_Devices">
        <Module_Header>
        </Module_Header>
        <Module_xxStandard_Log_Format>
        </Module_xxStandard_Log_Format>
        <Module_xxExtended_Log_Format>
        </Module_xxExtended_Log_Format>
    </Class>
    <Class Type = "Monitoring_Network">
        <Module_Header>
        </Module_Header>
        <Module_xxStandard_Log_Format>
        </Module_xxStandard_Log_Format>
        <Module_xxExtended_Log_Format>
        </Module_xxExtended_Log_Format>
    </Class>
    <Class Type = "Firewall">
        <Module_Header>
        </Module_Header>
```

```
            <Module_xxStandard_Log_Format>
            </Module_xxStandard_Log_Format>
            <Module_xxExtended_Log_Format>
            </Module_xxExtended_Log_Format>
        </Class>
        <Class Type = "IPTables">
            <Module_Header>
            </Module_Header>
            <Module_xxStandard_Log_Format>
            </Module_xxStandard_Log_Format>
            <Module_xxExtended_Log_Format>
            </Module_xxExtended_Log_Format>
        </Class>
        <Class Type = "Packet_Filter">
            <Module_Header>
            </Module_Header>
            <Module_xxStandard_Log_Format>
            </Module_xxStandard_Log_Format>
            <Module_xxExtended_Log_Format>
            </Module_xxExtended_Log_Format>
        </Class>
        <Class Type = "VPN">
            <Module_Header>
            </Module_Header>
            <Module_xxStandard_Log_Format>
            </Module_xxStandard_Log_Format>
            <Module_xxExtended_Log_Format>
            </Module_xxExtended_Log_Format>
        </Class>
    </Group>
</Format>
```